

# **For Reference**

---

**NOT TO BE TAKEN FROM THIS ROOM**



Ex libris  
UNIVERSITATIS  
ALBERTAENSIS









THE UNIVERSITY OF ALBERTA

DECISION TABLE PROCESSING OF THE CANADIAN  
STANDARDS ASSOCIATION SPECIFICATION S16.1

by



SIU KWONG FRITZ WU

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH  
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE  
OF MASTER OF SCIENCE

DEPARTMENT OF CIVIL ENGINEERING

EDMONTON, ALBERTA

SPRING, 1976



## ABSTRACT

This thesis presents sections of the Canadian Standards Association Specification 316.1 - Steel Structures for Buildings - Limit States Design, in decision table format. Basic theory of decision tables is discussed. The decision table formulation renders the code requirement checking procedure into an individual module independent from the analysis and member selection procedures.

TO

A specification processing program, in interactive mode and batch MY PARENTS, is presented.

A scheme for recursive execution of a decision table within a cycle has been developed leading to the possible saving of overall data items and decision tables.

A number of examples are given to check the validity of the decision tables compiled.





## ABSTRACT

This thesis presents sections of the Canadian Standards Association Specification S16.1 - Steel Structures for Buildings - Limit States Design, in decision table format. Basic theory of decision tables is discussed. The decision table formulation renders the code requirement checking procedure into an individual module independent from the analysis and member selection procedures.

A specification processing program, in interactive mode and batch mode, is also presented.

A scheme for recursive execution of a decision table within a cycle has been developed leading to the possible saving of overall data items and decision tables.

A number of examples are solved to check the validity of the decision tables compiled.





## ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to Prof. D.W. Murray, for his continuous encouragement, suggestions and patience in supervising the thesis.

Thanks are also extended to Prof. P.F. Adams and Prof. J.S. Kennedy for serving in the examining committee.

The expert typing of the manuscript by Mrs. Doreen Wyman is also greatly appreciated.



# TABLE OF CONTENTS

CHAPTER	PAGE
Dedication	i
Abstract	v
Acknowledgment	vi
Table of Contents	vii
List of Tables	xi
List of Figures	xii
I INTRODUCTION	1
1.1 Objective	1
1.2 The Design Process	1
II DECISION TABLE THEORY	4
2.1 Introduction	4
2.2 Basic Components	5
2.2.1 The Condition Stub	5
2.2.2 The Action Stub	5
2.2.3 The Condition Entries	5
2.2.4 The Action Entries	6
2.2.5 The Ingredients of Conditions and Actions	6
2.2.6 Rows and Rules of A Decision Table	6
2.3 Size of A Decision Table	7
2.4 Conventions	8
2.4.1 Conditions	8
2.4.2 Actions	8
2.4.3 Data Requirements	9
2.5 An Introductory Example	9





CHAPTER	PAGE
III DATA ORGANIZATION	11
3.1 Nature of Data Elements	11
3.2 Sources and Presence of Data Elements	11
3.3 Table Pointers and Mutually Exclusive Sets	13
3.4 Ingredients and Dependents of Data Elements	14
IV PROCESSING OF DECISION TABLES	17
4.1 Concepts	17
4.2 Conditional Execution	18
4.3 Direct Execution	19
V IMPLEMENTATION OF DECISION TABLE PROCESSOR	20
5.1 Computer Coding of Decision Tables	21
5.2 Storage Arrays for Decision Tables	23
5.3 Input Procedure for Decision Tables	24
5.4 Condition and Action Subroutines	25
VI DECISION TABLING THE CSA S16.1 - STEEL STRUCTURES FOR BUILDINGS - LIMIT STATES DESIGN	27
6.1 General Description of CSA S16 Standard	27
6.2 Scope of the Tables Compiled in This Thesis	28
6.3 Organization of The Decision Tables	28
VII THE SPECIFICATION PROCESSING PROGRAM	31
7.1 Processing Procedure - Batch Mode	31
7.2 Processing Procedure - Interactive Mode	33
7.3 Recycling Procedure	34





CHAPTER	PAGE
7.4 Clearing of Dependent Data	35
7.5 Recursive Use of Tables In A Cycle	36
VIII APPLICATION EXAMPLES	39
8.1 Example 1 - Axially Loaded Column	39
8.2 Example 2 - Laterally Unsupported Beam	42
8.3 Example 3 - Axial Compression and Bending	43
IX SUMMARY AND CONCLUSION	58
List of References	108
APPENDIX A HEIRARCHY CHARTS AND DECISION TABLE INDEX	111
APPENDIX B GLOBAL DATA SHEETS, DATA DEPENDENT LISTS, AND DECISION TABLES OF CSA S16.1	125
APPENDIX C USER'S GUIDE	204
Introduction	204
C.1 Program Limitations	204
C.2 Batch Mode Control Cards	205
C.3 Data Input for Batch Mode	205
C.4 Interactive Mode Control Commands	209
C.5 I.O. Units for Interactive Mode	210
C.5.1 Data in I/O Unit 5	210
C.5.2 Data In I/O Unit 9	211
C.5.3 Data In I/O Unit 2	211
C.5.4 Data In I/O Units 8,6,4,7	212
C.6 Description of MTS Files	212



APPENDIX D	INTERACTIVE MODE PROCESSING PROGRAM	220
	- Source Listing	
APPENDIX E	BATCH MODE PROCESSING PROGRAM	249
	- Source Listing	





## LIST OF TABLES

TABLE	TITLE	PAGE
1	Design Calculation for Example 1	46
2	Design Calculation for Example 1	47
3	Design Calculation for Example 2	48
4	Design Calculation for Example 2	49
5	Design Calculation for Example 2	50
6	Design Calculation for Example 3	51
7	Design Calculation for Example 3	52
8	Design Calculation for Example 3	53
9	Design Calculation for Example 3	54
10	Design Calculation for Example 3	55
11	Design Calculation for Example 3	56
12	Design Calculation for Example 3	56



## LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	Basic Components of a Decision Table	63
2.2	A Complete Decision Table	63
2.3	A Reduced Size Decision Table	64
2.4	Data Requirements of a Decision Table	64
3.1	Global Arrays	65
3.2	Generating Dependents from Ingredients of a Condition or an Action	66
3.3	Illustration of Dependence Concept by the Logic of Decision Tables	67
3.4	Flow Chart for Generating Dependents by the Logic of Decision Tables	68
4.1	Processing of a Decision Table	69
5.1	Permanent and Temporary Decision Table Storage Arrays	70
5.2	Permanent, Temporary Arrays and Their Pointer Arrays	71
5.3	Overall Array Structure and Interrelationship	72
5.4	Coded Input Form of Decision Table 13.5.A.1 (44)	73
5.5	Condition and Action Subroutines for Decision Table 13.5.A.1 (44)	74
7.1	Subroutine Structure for Batch Mode Processing	75
7.2	Missing Data Input Procedure in Interactive Mode	76
7.3	Procedure for Clearing Dependent Data Elements	77
7.4	Procedure for Clearing Dependent Data Elements	78



7.5	Recursive Operation of a Decision Table	79
8.1	Loading Condition for Example 1	80
8.2	Terminal Output for Example 1	81
8.3	Computer Output for Example 1	82
8.4	Computer Output for Example 1	83
8.5	Computer Output for Example 1	84
8.6	Computer Output for Example 1	85
8.7	Terminal Output for Example 2	86
8.8	Computer Output for Example 2	87
8.9	Computer Output for Example 2	88
8.10	Computer Output for Example 2	89
8.11	Computer Output for Example 2	90
8.12	Computer Output for Example 2	91
8.13	Computer Output for Example 2	92
8.14	Loading Condition for Example 3	93
8.15	Terminal Output for Example 3	94
8.16	Computer Output for Example 3	95
8.17	Computer Output for Example 3	96
8.18	Computer Output for Example 3	97
8.19	Computer Output for Example 3	98
8.20	Computer Output for Example 3	99
8.21	Computer Output for Example 3	100
8.22	Computer Output for Example 3	101
8.23	Computer Output for Example 3	102
8.24	Computer Output for Example 3	103
8.25	Computer Output for Example 3	104
8.26	Computer Output for Example 3	105
8.27	Computer Output for Example 3	106
9.1	The Complete Design Process	107
A.1	Decision Table Heirachy Chart 1	112
A.2	Decision Table Heirachy Chart 2	113
A.3	Decision Table Heirachy Chart 3	114
A.4	Decision Table Heirachy Chart 4	115
A.5	Decision Table Heirachy Chart 5	116





A.6	Decision Table Heirachy Chart 6	117
A.7	Decision Table Heirachy Chart 7	118
A.8	Decision Table Heirachy Chart 8	119
A.9	Decision Table Heirachy Chart 9	120
A.10	Decision Table Heirachy Chart 10	121
A.11	Decision Table Heirachy Chart 11	122
A.12	Explanation of Table Designation	123
A.13	Decision Table Index	124
C.1	Arrangement of Data Input Cards for Batch Mode	215
C.2	File OBCOMBINE	216
C.3	File OBCOMBINER	216
C.4	Data File LUB	217
C.5	Data File AXCOM	218
C.6	Data File COMBEN	219



# CHAPTER I

## INTRODUCTION

### 1.1 Objective

The purpose of this thesis is to document and implement the 1974 Canadian Standard Association (CSA) Standard S16.1-Steel Structures for Buildings-Limit States Design, in the form of decision tables. The technique of decision logic tables, or simply "decision tables", appears to be the best means available at present for clear definition and presentation of design logic.

It is hoped that this form of presentation will not only help to give a more precise and logical interpretation of the Standard but also facilitate an easier approach in implementing it for computer applications. Implementation of CSA S16.1 in this thesis is accomplished through a general purpose decision table processing program listed in Appendix D and E, which results in the ability to check the adequacy of most types of members against all relevant provisions of the Standard (also referred to as a code or specification).

### 1.2 The Design Process

The design process can be considered as consisting of the following steps:





- (i) Analysis
- (ii) Member Selection
- (iii) Constraints Checking (i.e. - checking all code provisions)

As the complexity of the code provisions increases, the more desirable a logical format becomes. Design specifications are generally a complex system in which all pertinent combinations of design events and their associated requirements are presented. Presentation of such logically complex systems in the traditional narrative format is difficult due to the necessity of sequential presentation of concepts and the limitations of language with respect to absolutely precise communication. Since design specifications essentially control the quality and safety of a structure, it is important that their requirements be explicit and clearly presented in order to facilitate their use to the best possible extent during the design process.

The common present approach of producing structural design programs based on a particular set of specifications is very inefficient. Specifications are often subjected to revisions. A minor revision often leads to a major modification of the program. In many cases, those responsible for the actual design of a structure do not know and often have difficulty in locating a person who knows exactly what assumptions



and interpretations have been included in the program. Hence, developing and revising such programs becomes a very expensive and time consuming task.

Most of the specifications and codes of different authorities at present are not readily adaptable to computer application because they do not display the logic leading to a certain action as clearly as needed for computer application, and the requirements for the same action may be scattered throughout the text of the specification. The design process up to the end of step (ii) involves many considerations other than design constraints whereas step (iii) is entirely dependent on them. Therefore, a more logical approach is to regard the constraints checking as a separate module, independent from analysis and member selection, and then design the programs accordingly. In this way, any member selection programs can be used with programs of different code requirements.

The decision table format of CSA-S16.1 presented in this thesis could therefore, result in a reassessment of the form in which this code is presented and may ultimately result in an improvement to the specification.



## CHAPTER II

### DECISION TABLE THEORY

#### 2.1 Introduction

Tables are a familiar part of everyone's life. From mathematical tables to the milage tables on road maps. They provide us with an orderly presentation of data.

Traditionally, flow charts and narratives have been used to structure the logic of a problem. These are effective as long as the problem is fairly simple logically. When interaction among a large number of conditions exists, how can one be certain that every possible combination has been considered?

Decision tables are simply a technique for recording "decision-making" processes. They have been used since the early 1950's. Unlike flow charts, which could be as simple or as detailed as the programmer desires, decision tables force one to be thorough and concise. When properly used, they demand that all combinations of conditions be considered and allow irrelevant tests to be deleted. In this chapter, the basic components and theory of decision tables are discussed. For a greater insight into the theory, the reader is referred to references: 25, 10, 14, 18, 20, 26, 22, 6.





## 2.2 Basic Components

A decision table is simply a tabular display of all elements of a segment of a problem. The table shows all of the conditions affecting the problem and their interrelationship. It further indicates which action or actions are appropriate for each set of conditions.

A decision table consists of sections, which are arranged as shown in Fig. 2.1\*. The contents of these sections are discussed below.

### 2.2.1 The Condition Stub

This area contains all the conditions relevant to the particular segment of the problem. They must all be "logical" conditions, which can have only two possible values: "YES" or "NO".

### 2.2.2 The Action Stub

This area contains all the possible actions that may be taken resulting from different combinations of conditions. The actions may be printing of messages, computations, or execution of other tables.

### 2.2.3 The Condition Entries

The questions asked in the condition stub are answered here. This segment lists all the pertinent

---

\* All Figures are grouped together preceding the References.



combinations of the logical conditions in a column. Each column specifies a "rule". Responses to conditions are restricted to "Y" for Yes, "N" for No, and "I" for Immaterial.

#### 2.2.4 The Action Entries

The elements of the action entries may either be "Y" or "blank". "Y" signifies the corresponding action in the action stub is to be performed. A "blank" signifies that the corresponding action is not to be taken.

#### 2.2.5 The Ingredients of Conditions and Actions

These two areas list the ingredients of each condition and action respectively. Ingredients are defined in Section 3.4.

#### 2.2.6 Rows and Rules of a Decision Table

Horizontal levels, called "rows", run across the entire table. The condition and action entry portions of the table are further subdivided into vertical columns called "rules".

Decision tables of the type described above are called "limited entry tables" because the condition entries are considered to be limited to "Y", "N" or "blank" and the action entries are limited to "Y" and "blank". Other types of decision tables such as



"extended entry tables" and "mixed entry tables" (25, 10, 14, 18, 20, 26), allow for a wider variety of condition and action entries but they are more complex to program.

### 2.3 Size Of A Decision Table

As explained in Sect. 2.2.1, each logical condition has only two values ("Y" or "N"). It follows that for a table with one condition, there are two rules. Therefore, a table with  $n$  conditions has  $2^n$  rules. A table of this type is termed a "Complete Table". However, a "Complete Table" such as that illustrated in Fig. 2.2, rarely occur in practice, for the following reasons:

- (a) If the action for rule 2 and 3 is the same, the value of condition 2 for rule 2 and 3 becomes immaterial. The table transforms into the form illustrated in Fig. 2.3. The number of rules has been reduced to 3.
- (b) In decision table X.1(1)\* of Appendix B, the two conditions are mutually exclusive as discussed in Sect. 3.3. Therefore, there are only two valid combinations of the two conditions, namely, Y N and N Y, rather than  $2^2 = 4$  number of rules in a complete table.

---

\* Decision tables in Appendix B are arranged, as closely as possible, in order of the number which appears in brackets.





## 2.4 Conventions

In this section, the notations and conventions used in the formulation of the decision tables presented in Appendix B are discussed. Signs and units of variables are the same as those used in CSA-S16.1.

### 2.4.1 Conditions

All the conditions in the condition stub are logical variables. There are two types of conditions:

- (i) Conditions which are logical data items, such as "SECTION DOUBLY SYMMETRIC?"
- (ii) Logical conditions which result from numerical calculations, such as " $b/t \leq 420/\sqrt{F_y}$ ?", which has a value of "YES" or "NO" after the calculations for  $b/t$  and  $420/\sqrt{F_y}$  are performed.

### 2.4.2 Actions

Certain actions which are listed in the action stub, are performed according to specific combinations of conditions in the condition stub. There are three types of actions:

- (i) An action which directs the execution of the program to another table. For example "Execute Table M".
- (ii) An action which outputs appropriate messages. For example, "STRENGTH CRITERION NOT SATISFIED".



- (iii) An action which calculates numerical values of variables. For example " $M_r = \phi z F_y$ ".

#### 2.4.3 Data Requirements

There are in general two types of input data required to process a decision table. The first type is conventional numerical data, such as the value of yield stress  $F_y = 44$  ksi. The second type is logical data, which can only have value of "YES" or "NO", such as "STRUCTURAL STEEL MEMBER?"

The data items necessary for the processing of each table are listed preceding each decision table, as presented in Appendix B, in the format shown in Fig. 2.4. The first column identifies the data item. The second and third columns specify whether this data item has to be externally input or can be obtained from another table.

#### 2.5 An Introductory Example

To demonstrate the use of a decision table, the Decision Table 13.4.A.1 (75) in Appendix B, for section 13.4 of CSA S16.1 Standard, which deals with the calculation of the factored shear resistance  $V_r$ , will be used. In this example, assume the problem is to calculate the factored shear resistance  $V_r$  of a beam which has been designed using plastic analysis. The values of the five conditions in the condition stub



that describe the problem are N Y Y N N. Scanning the condition entry part of the table, rule 4 satisfies the condition requirements, since conditions 4 and 5 of rule 4 are both "immaterial". Following rule 4 to the action entry part of the table, it is noted that there is a Y entry opposite action 4, which means action 4 in the action stub will be performed. It is a numerical calculation which gives a value of  $V_r = 0.55\phi_w d F_y$ .





## CHAPTER III

### DATA ORGANIZATION

This chapter discusses the nature, properties and sources of values of data elements. In a general processor program, the number of data values required becomes very large even for a small problem. Therefore, it is imperative to have a procedure which can handle large amounts of data efficiently. In other words, it largely becomes a data management problem (13, 19).

#### 3.1 Nature Of Data Elements

There are in general, two types of data elements:

- (a) Data which have numerical values in the conventional sense. For example, the length of the steel member, l.
- (b) Data which are logical in nature. They can only have values of "YES" or "NO". For example, "Is the section double symmetric?".

#### 3.2 Sources and Presence of Data Elements

There are four sources from which data values can be generated:

- (a) Data which are supplied by external input. They can either be logical or numerical.



- (b) Data which are generated by actions performed in a decision table. Data generated in this manner can either be logical or numerical.
- (c) Data which are generated as a result of calculations performed on other data elements. For example, " $b/t \geq 420/\sqrt{F_y}$  ?". This type of data element is always logical in nature.
- (d) Data to be input from the terminal using Subroutine READIN. This source is only applicable in interactive mode operation which is discussed in Sect. 7.2.

The sources of data discussed above are listed in the hierarchy sequence in which the program searches for data. During processing, when a particular data value is required, the program checks if it has been input externally. If not, it then turns to sources (b), (c) and (d) in that order.

It should be noted that not all the actions of a decision table generate data elements. However, the values of all the data elements generated may be stored in a global array called DATA, together with the values of each data element from the condition and action stubs. The value of each data element may be retrieved from the array DATA by using its number address (i.e. subscript) in the array. This arrangement of global storage enables data to be retrieved easily at every stage during execution of the program.



Due to the multiple sources from which data elements can be generated, it becomes essential to know at any time during processing whether a particular data item has a value or not. This information is provided by an array PRD (Presence of Data), whose subscripts correspond to those of array DATA, and which contains logical flags `.TRUE.` or `.FALSE.` . The elements in array PRD are all initialized to `.FALSE.` at the beginning of execution of the program. Whenever the value of a data element becomes available during execution, the corresponding element in array PRD is set to `.TRUE.` . The presence of a data element value can then be established at any time during processing by checking with array PRD.

### 3.3 Table Pointers and Mutually Exclusive Sets

In order to obtain a value of a particular data element by the execution of an appropriate decision table, according to the data source of Sect. 3.2(b), the number of the table which must be executed is specified as a property of that data element. This table number is stored in an array TABD (Table for Data) with subscripts corresponding to those in the array DATA for each data element. A blank in TABD indicates that the data element cannot be retrieved by executing any decision table.



Some data elements have the property that their values are mutually exclusive from one another, forming a mutually exclusive set. Data in a mutually exclusive set are always logical in nature. The implication of a mutually exclusive set is the user need only supply the value of one item in the set. The other items in the same set will automatically be set to "NO". For example, in Decision Table 11.A (6) in Appendix B, the list of data required consists of the following: T-Section, I-Section, J-Section, Rectangular Hollow Section and Box Section. These five data items form a mutually exclusive set since only one can have a value of .TRUE. at any given time. The number of a mutually exclusive set to which a data element belongs is stored in an array ISET. The property of mutually exclusive sets discussed above reduces the number of externally input data considerably.

The subscript (or address) of a data item in the array DATA also serves to retrieve the values in arrays PRD, TABD, and ISET which are applicable to this data item. The interrelationship of these four global arrays is as illustrated in Fig. 3.1.

### 3.4 Ingredients and Dependents of Data Elements

As indicated in Sect. 3.2(b,c), there are two ways in which data elements can be generated by involving





other data elements. When a value of a data element is evaluated as a function of other data elements, the latter are called "ingredients" of the former. Thus, a data element  $X$  which may be evaluated as  $X = F(u_1, u_2, u_3 \dots u_n)$ , has data elements  $u_1, u_2, u_3 \dots u_n$  as its ingredients. For example, the logical data element " $b/t > 420/\sqrt{F_y}$ ?" has ingredients  $b, t$ , and  $F_y$ .

There are instances when the value of one data element may be generated by more than one function, each having a different set of ingredients. For example, the data element  $M_{ry}$  in Decision Table 13.6.A.2 (50) in Appendix B, may be generated by three different actions which have different ingredients. Since ingredience is a property of the data element, it is only necessary to associate ingredients with the conditions and actions of decision tables where the ingredients are actually used. The dependent list of each data element is generated internally. Thus, ingredients are associated with rows of a decision table.

The concept of dependence follows directly from that of ingredients. If  $X$  has  $Y$  as its ingredient, then  $X$  is a dependent of  $Y$ . In the above example the logical data element " $b/t > 420/\sqrt{F_y}$ ?" is a dependent of  $b, t$  and  $F_y$  because if any of these change in value, the logical data element may also change in value. The internal procedure of generating dependents imposed by



ingredients consists of a loop for the data element which involves a search to see if the data element has any ingredients. If yes, it is placed in the lists of dependents of all the ingredients, unless it is already there. This procedure is illustrated by the flow chart in Fig. 3.2.

Dependence of data elements can also be imposed by the logic of decision tables. Actions performed in the action stub due to a particular combination of conditions are dependents of those conditions, as illustrated in Fig. 3.3, where data elements produced by action B are dependents of conditions A and B. Dependents generated by the logic of decision tables are entered into the lists of dependents of the relevant conditions in the condition stub by a loop in Subroutine SETUP, as flow charted in Fig. 3.4. It should be noted that dependency is a function of the ingredients and the decision logic of the immediate decision table. Dependency beyond this level need not be explicitly established as a data property but is determined at execution time, when necessary for clearing purposes, by the processing program. This procedure is discussed in Sect. 7.4.



## CHAPTER IV

### PROCESSING OF DECISION TABLES

#### 4.1 Concepts

The approach taken to process the decision tables in Appendix B is to express each decision table as part of the global data, and execute all the tables by a single program. The procedure for processing a complete set of interrelated tables consists of the sequential processing of single tables in the order required by the problem, and checking the availability of data for each of these tables. If a data element is not available, execution of that table is suspended while the data element is being retrieved according to the heirarchy of data sources discussed in Sect. 3.2. This gives rise to the concept of conditional and direct execution which is discussed in Sect. 4.2 and 4.3.

In this section, the method of processing a decision table is introduced. The procedure is flow charted in Fig. 4.1. The data values required are assumed to be all available. The program starts by checking to see if the data values of the conditions match the condition entries of the first rule of the table. Matching is skipped if the condition entry is immaterial. Each rule is tested in turn until a match is found between the data values of the conditions and





all condition entries for a rule. The relevant actions in the action stub of the matched rule will then be performed. If no rule is found to match the given conditions, the program will take an error exit route and gives a message to that effect.

In general, not all the data elements need be externally input. If a data value is missing, be it a condition or an action, it can be obtained by the procedure described in Sect. 4.2.

#### 4.2 Conditional Execution

There are three situations which will lead to the suspension of execution of a table:

- (a) The value of a condition is missing
- (b) The ingredients of a condition are missing
- (c) The ingredients of an action are missing.

In all of the above cases, if a table address is given in the array TABD (Fig. 3.1) for the missing data element, the execution of that table will commence immediately while the current table is suspended. The reason of suspension (a, b or c above), the current rule number, the current condition or action number and the current ingredient number of the suspended table are stored for re-execution. Once the missing data has been obtained, control is returned to the suspended table.



If a table address is not given in the array TABD for the missing data element, then the program checks to see if it has ingredients. If the missing data element has ingredients (and they are all present), a "condition subroutine" or "action subroutine" is called to evaluate the element. These subroutines are discussed in Sect. 5.4. If the data element does not have an entry in TABD and also has no ingredients, execution is terminated if in batch mode. In interactive mode, the subroutine READIN is called as discussed in Sect. 7.2.

#### 4.3 Direct Execution

The command for direct execution is "Execute Table X". This is an action specified in the action stub and is referenced by a different code in the action entries. Once execution of Table X is completed, control is returned to the next action of the original table. Unlike conditional execution, direct execution may only occur as a result of an instruction in the action entry of a rule.



## CHAPTER V

### IMPLEMENTATION OF DECISION TABLE PROCESSOR

The decision table processing program used herein was originally developed by Goel (12) in FORTRAN IV with batch-mode only input facility. The source program listed in Appendix E is similar to the original program but with numerous modifications implemented by the writer. In order to increase the capability of the processor, an interactive-mode of input and execution, and a recursive execution scheme have been introduced which are discussed in Chapter VII.

For specification checking, the program requires two types of data input:

- (a) Decision table information which is independent from any other data characteristics. Therefore, decision tables only need to be processed once and stored permanently unless alterations are made to the tables. The storage of this type of input is discussed in Sect. 5.2, and the input procedure is discussed in Appendix C.
- (b) Problem orientated data values which are not part of permanent data. For example, "The depth of section d" or "The loadings on the member", as discussed in Sect. 3.2.



As discussed in Chapter III, data elements are stored in global arrays (Fig. 3.1). Therefore each data element can be easily retrieved by its array subscript. For example, DATA (95). However, it is often more convenient and meaningful to identify a data element with a name. This can be achieved by using the FORTRAN EQUIVALENCE STATEMENT. For example, "EQUIVALENCE (DATA (95), \$CLAS1)". In this case, data element 95 in array DATA is a logical element \$CLAS1= "IS SECTION A CLASS 1 SECTION?". Throughout this thesis, logical data elements are identified by the prefix character "\$".

### 5.1 Computer Coding of Decision Tables

The basic components of a decision table have been discussed in Sect. 2.2. In this section, the coding of each section of a decision table in a form suitable for input to the computer is described.

- (a) The condition stub contains logical data elements whose values are stored in the global array DATA. This area contains the addresses (i.e. subscripts) of the conditions in the array DATA.
- (b) The action stub contains both numerical and logical data elements. Three types of actions may be performed:
  - (i) Calculation to obtain the value of a data element by an action subroutine. The action





stub contains the address of the data element in array DATA in this case.

- (ii) Direct execution of a table. The action stub contains the table number in this case.
- (iii) Output of messages by an action subroutine.  
The action stub contains a "blank" for this type of action.
- (c) The condition entry portion of the table has three types of entries. They are "Y", "N" and "I". A "zero" or a "blank" signifies an immaterial entry. A "1" and "2" signifies a YES and NO entry respectively.
- (d) The coding in the action entry portion of the tables signifies which type of action discussed in (b) is to be performed. The following coding is used:  
A "blank" signifies that no action is to be taken.  
A "1" signifies that the action is to be performed by an action subroutine (Sect. 5.4). The action may be calculating a data element or printing out of messages.  
A "2" indicates direct execution of another table.
- (e) The ingredient list for a condition contains the addresses in array DATA of the ingredients. The maximum number of ingredients per condition which the program can handle is eleven.
- (f) The ingredient list for actions is similar to that for conditions.



## 5.2 Storage Arrays for Decision Tables

Economy in primary storage and efficient access to it are of particular importance in a program which handles large volumes of data information (13, 19). In order to conserve storage, the size of each decision table is defined at the time of input. The size of a table is defined by the number of rules, the number of actions, and the number of conditions. Each of these parameters is stored in the single subscripted arrays L, M, and N respectively. The contents of each decision table are read initially into temporary arrays and then compacted into the permanent one-dimensional arrays LARRY1 to LARRY6, as shown schematically in Fig. 5.1.

There is a pointer array assigned to each permanent array for the purpose of locating data elements efficiently. Fig. 5.2 shows a tabulation of the permanent arrays and their corresponding pointer arrays. As a typical example, the pointer array IPNTRC stores the base address (ie. the reference subscript) of each condition for the purpose of locating ingredients of conditions. The  $J^{\text{th}}$  ingredient of the  $I^{\text{th}}$  condition in array LARRY1 can be obtained by:

$$I1 = \text{IPNTRC}(I) + J$$

$$\text{IDATA} = \text{LARRY5}(I1)$$

IDATA is the data address in array DATA of the  $J^{\text{th}}$  ingredient. The number of ingredients which condition I has is obtained from:



IPNTRC(I+1) - IPNTRC(I)

Fig. 5.3 shows schematically the overall array structure and their interrelationship.

### 5.3 Input Procedure for Decision Tables

For the purpose of input, the decision tables need not be modified in form to accommodate the storage scheme described in Sect. 5.2. The construction of the temporary arrays and the one-dimensional permanent arrays is entirely an internal operation.

The tables are read in one after another but need not be in any particular sequence since each table is identified by a table number. The first card for each table is a table header card which contains the table number (T), the number of rules (L), the number of actions (M) and the number of conditions (N). The header card is followed by a card for each row of the decision table, first the condition rows, then the action rows. Each of these cards contain the data address of the condition or action stub, the condition or action entries code, the data addresses of the condition or action ingredients and a flag "C" if more ingredients are to follow on the next card. A maximum of eleven ingredients per condition or action is permitted. Fig. 5.4 illustrates the coded input form of Decision Table 13.5.A.1 (44) in Appendix B. The input format is discussed in Appendix C: User's Guide.



#### 5.4 Condition and Action Subroutines

These are subroutines associated with the condition or action stub of each decision table. The function of the condition subroutine is to calculate logical values of conditions whereas the action subroutine is used for the calculation of numerical values of data elements or for the output of messages. For example, data sources (b) and (c) discussed in Sect. 3.2 require the use of such subroutines.

Each subroutine is accessed from the main routine by a computed GO TO statement where the integer table number T controls program flow to the proper call statement as shown in the main routine in Appendix D. The presence of the ingredients which are required by each subroutine is checked before executing. The only action which does not require any ingredients is the output of messages. If any ingredients are missing, conditional execution, as discussed in Sect. 4.2, is activated to retrieve the missing ingredients. The actual portion of the subroutine relevant to a particular condition or action is referenced by a computed GO TO statement within the subroutine where the condition or action number controls program flow. A typical example of condition and action subroutines is presented in Fig. 5.5. This figure illustrates the subroutines for Decision Table 13.5.A.1 (44) in Appendix B, which





calculates the moment of resistance  $M_{rx}$  and the ratio  $R2$ .  
Condition subroutines are designated by CC and action  
subroutines by AA, followed by the table number T.



CHAPTER VI  
DECISION TABLING THE  
CSA S16.1 STANDARD - STEEL STRUCTURES FOR BUILDINGS -  
LIMIT STATES DESIGN

6.1 General Description of CSA S16 Standards

The CSA S16 Standard is the principal guide for the design of steel buildings in Canada. The first CSA S16 Standard appeared in 1924. The 1969 edition (15) of the standard was revised in 1974. The new Standard, CSA S16.1-1974 - Steel Structures for Buildings - Limit States Design, provides rules, guidelines and requirements for the design, fabrication and erection of steel buildings where the design is based on limit states (2). This Standard takes its place beside S16-1969, which will continue to provide engineers with a working stress design standard for some time.

The limitations in precisely wording the texts of standards often leads to difficulties in interpretation and possible inconsistencies in their application.

The objective of this thesis is to decision table the CSA S16.1 and incorporate the tables into a general purpose processing program for checking a design



against the Standard. The Standard was still in the drafting stage when the tables were being compiled and the decision tables are based on the October 1974 draft.

## 6.2 Scope Of The Tables Compiled In This Thesis

The decision tables presented in Appendix B cover the sections in Chapters 1 to 13, Chapter 15 and Chapter 19 of the CSA S16.1 Standard, which deal with design decisions and which are not purely descriptive in nature. Regrettably due to the amount of time available, the other chapters of the Standard could not be compiled.

The tables and the Standard, do not provide certain computational procedures such as computing member loadings and forces, or methods of computing section properties. These procedures must be provided by the user and input to the program as external input data. Decision tables are capable of developing all the possible combinations of conditions. However, certain combinations may be invalid or impractical. Therefore, in the development of the decision tables in this thesis, only those combinations which are relevant were incorporated into them.

## 6.3 Organization Of The Decision Tables

In Appendix A, heirachy charts of the tables compiled in this thesis are presented. These charts show the order of execution of each table when a particular checking task is performed.



Figs. A.1 to A.11 show the general outline of the table organization. A description of the table designation is given as an introduction to the Appendix. Decision Table X.1(1) makes the decision whether checking is to follow the elastic or plastic analysis procedure. The former route is treated in Decision Table X.2(2) and the latter route in Decision Table 8.5 (78).

The elastic analysis procedure treats structural elements according to one of the following.

- (a) Structural Steel Member
- (b) Girder
- (c) Bearing Stiffener
- (d) Intermediate Transverse Stiffener
- (e) Connection
- (f) Built-up Member
- (g) Composite Member
- (h) Open-Web Steel Joist

Structural steel members are dealt with starting from Decision Table X.3(3), and the chain of execution of the subsequent decision tables are as shown in Figs. A.2 to A.9. Fig. A.2 deals with axial compression or axial tension stresses, shear stresses and bending stresses. Figs. A.3 to A.4 deal with the classification of sections. Figs. A.5 to A.7 deal with pure bending stresses. Finally, Figs. A.8 and A.9 deal with combined stresses.





Fig. A.10 displays the chains of execution for girders, bearing stiffeners and intermediate transverse stiffeners. Fig. A.11 displays the tables required for checking of designs using plastic analysis.

In these heirarchy charts, the full lines represent direct execution and the broken lines represent conditional execution. Sections of the Standard dealing with connections, built-up members, composite members, and open-web steel joists have not been decision tabled.



## CHAPTER VII

### THE SPECIFICATION PROCESSING PROGRAM

This chapter discusses the various operational modes of the processing program used in checking the constraints of the Standard, CSA-S16.1. The program will also be able to check constraints imposed by other standards (or codes), so long as they are presented in decision table form. The original program as developed by GOEL (12), limits execution to batch-mode only. It is also only capable of executing a given decision table once in a particular cycle. Therefore, the versatility of the processor is severely handicapped. Modified batch mode, interactive mode and recursive execution of a table within the same cycle are introduced in this chapter.

The processing program, both in batch mode and interactive mode, has been implemented on the IBM/360/67 at the University of Alberta computing centre which operates under the Michigan Terminal System (MTS).

#### 7.1 Processing Procedure - Batch Mode

This section discusses the batch mode operation. The source program for this procedure is listed in Appendix E. The program in batch mode consists of the following subroutines, whose functions are described



below. Their interrelationship and sequence of execution is illustrated schematically in Fig. 7.1

- (a) MAIN Routine
- (b) SETUP Subroutine
- (c) INITIAL Subroutine
- (d) INPUT Subroutine
- (e) SETS Subroutine
- (f) STAK Subroutine
- (g) OUTPUT Subroutine

The MAIN Routine does all the constraints checking. The objective when checking design constraints is to identify the applicable rule in each decision table according to a given combination of conditions. This involves matching the condition stub with the corresponding condition entries. Before matching, all required data elements are checked for their availability. In batch mode, if a data element is not available from the data source discussed in Sect. 3.2, the program will take an error exit route aborting the run.

Subroutine SETUP reads the decision table input, the properties of data and creates temporary and compacted permanent arrays. The dependents of data elements are also generated by this subroutine.

Subroutine INITIAL which is called by subroutine SETUP, initialized all the arrays needed by the program.



Subroutine INPUT is used for reading the externally input data and for clearing of dependent data for the second or subsequent cycles. This subroutine is called by the MAIN ROUTINE.

Subroutine SETS is used to evaluate the data in mutually exclusive sets at the time of external input and is called by subroutine INPUT.

Subroutine STAK performs the stacking of decision tables for conditional execution and generates messages to that effect.

Subroutine OUTPUT outputs all the data elements which have a value at the end of each cycle for the purpose of checking and diagnosis.

## 7.2 Processing Procedure - Interactive Mode

This section discusses the procedure in implementing the interactive mode. In batch mode operation as discussed in the last section, control of the program takes the error exit route, terminating execution whenever a data element is missing and cannot be evaluated internally. This results in a large number of abortive runs which is both uneconomical and time (real time and CPU time) consuming. The interactive mode is implemented with these inefficiencies in mind. The main advantage of this mode is that the user can input data items from the on-line terminal, as an alternative to termination of execution.





Instead of taking the error exit route for the reasons discussed above, the subroutine READIN is called by the MAIN ROUTINE to input the missing data element from the terminal. A message is printed on the terminal indicating the nature of the element. The main steps of the procedure are as follows and are also illustrated in Fig. 7.2.

- (a) Read in missing data item (its subscript and value). Input a negative interger value for the subscript if the user wishes to terminate execution.
- (b) Call subroutine SETS if the data item belongs to a mutually exclusive set.
- (c) Clear the data item's dependents if not in the first cycle.
- (d) Return to MAIN ROUTINE.

Illustrative worked examples and their computer output are presented in Chapter VIII. A detailed input procedure is also provided in the USER'S GUIDE in Appendix C. The source program listing of interactive mode procedure is presented in Appendix D.

### 7.3 Recycling Procedure

In a design problem, the checking of a number of design alternatives may be required before a final design is chosen. For this reason, a recycling facility



has been incorporated into the processor with the interactive mode procedure. After the completion of cycle one execution, the program awaits the value of the variable INDIC from the terminal. A value of 2 for INDIC indicates there are no further cycles, whereas a value of 1 indicates a further cycle is required. The external data required for this next cycle can be input directly from the terminal. The limit of the number of cycles per run is 999.

In the second and subsequent cycles, a large number of data will be unchanged. Usually only a limited number of data elements will require alteration from the previous cycle. The clearing of dependents of the altered data elements is done internally in subroutine INPUT by setting their "flags" in the array PRD to .FALSE. . This clearing procedure is discussed in Sect. 7.4.

#### 7.4 Clearing Of Dependent Data

The characteristics of ingredients and dependents discussed in Sect. 3.4 lead to the conclusion that when a data element changes its value, its dependents, if any, will change their values as well. Hence it becomes necessary to clear the dependents of a data element once it has an altered value. Clearing of a data element is done simply by setting its presence flag in array PRD to .FALSE. . From the ingredience-dependence relationship



discussed, it can be deduced that it is possible for dependents to have their own dependents, down to a number of levels. During the dependent clearing process, these dependents of dependents will also have to be cleared. A data element stacking procedure is provided by the Subroutine STAK with the clearing process which is flow charted in Figs. 7.3 and 7.4.

### 7.5 Recursive Use Of Tables In A Cycle

There are numerous situations where it would be advantageous to execute a decision table more than once. For example, to calculate the class of several compression elements of a built-up member.

In the original program (12), such checking either had to be performed in separate runs or in different cycles.

Fig. 7.5 illustrates a scheme whereby decision tables can be written in such a way that they can be used recursively in one cycle with the existing processing program. The decision tables associated with section 13.8 of the Code, which checks the constraints for axial compression and bending, have been rewritten as decision tables 56R to 59R as shown in Appendix B, in order to illustrate the application and potential of this scheme. The scheme makes use of counters CHECKI and CHECKN to keep track of the number of times the decision table has been executed.



The steps of checking a Class 1, I-Section for axial compression and bending, as carried out by decision tables 56R to 59R, and as illustrated in Fig. 7.5, are as follows.

- (a) Table 56R determines whether the problem is axial compression and bending or axial tension and bending. The value of counter CHECKI is initialized to zero.
- (b) Since the section is a class 1, I-section. Rule number 1 applies in Table 57R, and the number of equations to be checked is 3. Hence, the value of CHECKN is set to 3, and Table 58R is to be directly executed.
- (c) After the actions in Table 58R are completed, control returns to Table 57R and the next action is to check whether  $CHECKI > CHECKN$ . If not, CHECKI is again increased by one and the checking cycle is repeated. If yes, control returns to the MAIN ROUTINE.

In Decision Table 57R, note that the data elements specified in the data statement DATA MCLEAR/, and their dependents must be cleared by subroutine CLEAR before the table is recycled. Otherwise these data elements will still be present from the previous cycle and the program will not recognize that the data elements must be recomputed.





From Decision Table 58 and Decision Table 58R, the number of equations needed to be checked for axial compression and bending is reduced from three to two by the scheme discussed above. Thereby, reducing the number of overall data elements. Consequently, the total number of decision tables required should be reduced. This is of advantage since in a standard (or code) of reasonable size, the number of decision tables and data elements will be very large. Reduction in their quantity will lead to shorter processing time and reduced storage requirements.



## CHAPTER VIII

### APPLICATION EXAMPLES

Three examples are solved in this chapter to illustrate the use of the processor described in chapters V and VII in checking sections under different combinations of stress resultants. The data requirement, a trace of the execution of the decision tables and the relevant computer output (both from the on-line terminal and the line printer) of each example are presented together with supporting manual design calculations to verify the computer results.

#### 8.1 Example 1 - Axially Loaded Column

Consider the column shown in Fig. 8.1 to be part of a frame-shear wall system. The 12 ft. column consists of two  $8 \times 6 \times \frac{5}{8}$  angles of G40.12 - 44W steel, supporting a maximum dead load of 300 kips and a maximum axial live load of 85 kips.

This example illustrates a typical program run in interactive mode. The output from the on-line terminal is presented in Fig. 8.2. The output from the line printer is presented in Figs. 8.3, 8.4, 8.5, 8.6. The bulk of the data information in the output are printed by the line-printer. In Fig. 8.2, statements in lower case letters are those input by the user, whereas statements



in upper case letters are those output by the computer. The MTS (Michigan Terminal System) files, input and output units associate with the first statement (\$RUN control command) are discussed in Appendix C: Users' Guide.

Since data items 222<sup>†</sup> and 223<sup>†</sup> are not available, and are not obtainable by executing any other tables, messages are then output on the terminal indicating the condition or action number of the table to which each data item corresponds. Subroutine READIN is called enabling the missing data items to be input directly from the terminal. Terminal data input can be of the free-format type which is of considerable advantage. The data item numbers (222 and 223) are input first followed by their values (0.0 in both cases).

Upon completion of checking, the design message "Strength Criterion Satisfied" is output on the terminal which indicates that the section is satisfactory for the imposed loading. At this stage, the execution of the program either terminates or re-executes another cycle depending on whether the variable INDIC is given a value of 1 or 2. In this example, a value of 2 is input from the terminal since no further cycles are required.

---

<sup>†</sup>For data description, see Appendix B.



Fig. 8.3 lists the externally input data items for cycle 1. "KGLOBAL" is the data number and "DATAK" is the value. These data items are read by Subroutine INPUT. When reading a data item, the subroutine simultaneously checks for whether it belongs to a mutually exclusive set. If so, the values of the other data items in the same set are set to 0.0. Fig. 8.4 lists the externally input data items together with their mutually exclusive set companions, if any, for the purpose of echo checking.

Fig. 8.5 shows a trace of all the decision tables which have been executed in cycle 1. Whenever the execution of a particular table is suspended, the reason and point of suspension are shown on this trace. Once the execution of each table is completed, the applicable rule number is indicated.

Finally, a design message "Strength Criterion Satisfied" is output indicating the section is satisfactory when supporting an axial compressive load of 385 kips. This message is also output on the terminal as shown in Fig. 8.2.

Fig. 8.6 lists the data items which have a value at the end of cycle 1. The correctness of their values is verified by the manual design calculations in Tables 1 and 2.





The significant data item in Fig. 8.6 is data number 150 which represent the ratio  $R1 = P_f/P_{rc}$ , where  $P_f$  is the factored axial compressive force and  $P_{rc}$  is the factored compressive resistance. The value of  $R1 = 0.8944$  ( $< 1.0$ ) indicates the section is satisfactory and not overdesigned. Hence a second cycle to choose a better section will not be necessary.

## 8.2 Example 2 - Laterally Unsupported Beam

A W12x50 wide - flange section, of G40.12-44W steel, is to be checked for suitability as a beam of length 24 feet. The section is laterally unsupported except at the ends. This beam is required to support a uniformly distributed dead load of 1.2 kip per foot, and a live load of 0.6 kip per foot.

This example is executed in interactive mode using the same procedure employed in Example 1. Figs. 8.7, 8.8, 8.9, 8.10, 8.11, 8.12, 8.13 present the computer output associated with this example. The content and significance of these figures are similar to those of Example 1 which have been discussed in detail with that example.

In Fig. 8.13, data number 151 which represents the ration  $R2 = M_{fx}/M_{rx}$ , where  $M_{fx}$  is the factored moment about the major axis and  $M_{rx}$  is the factored moment of resistance about the major axis, has a value of 0.9724



which indicates the section has sufficient moment capacity. The ratio  $V_f/V_r$ , where  $V_f$  is the factored shear force and  $V_r$  is the factored shear resistance, has a value of 0,24 which indicates the section is satisfactory in shear. The correctness of the data values in Fig. 8.13 is verified by the manual design calculations in Tables 3 to 5.

### 8.3 Example 3 - Axial Compression and Bending

A W10x49 wide-flange section, of G40.12-44W steel, is to be checked for suitability as a column in a building with a 9.5 foot storey height. The sway effects due to wind and other lateral loads are to be resisted by a bracing system. The dead and live load moments, and the axial compressive load on the member are shown in Fig. 8.14.

Figs. 8.15, 8.16, 8.17, 8.18, 8.19, 8.20, 8.21 present the computer output for cycle 1 checking. Upon the completion of this cycle, the section is found to be unsatisfactory both in strength and stability requirements as indicated by the message in Fig. 8.20. From the data values output at the end of cycle 1, as shown in Fig. 8.21, the values of the following three equations are found to be:



$$(i) \quad \frac{M_{fx}}{M_{rx1}} + \frac{M_{fy}}{M_{ry1}} = 1.0017 (> 1.0)$$

$$(ii) \quad \frac{P_f}{P_{rc2}} + \frac{0.85 M_{fx}}{M_{rx1}} + \frac{0.6 M_{fy}}{M_{ry1}} = 0.9888 (< 1.0)$$

$$(iii) \quad \frac{P_f}{P_{rc1}} + \frac{\omega_x M_{fx}}{M_{rx2}(1 - P_f/c_{ex})} + \frac{\omega_y M_{fy}}{M_{ry1}(1 - P_f/c_{ey})}$$

$$= 1.0916 (> 1.0)$$

These values are verified by the manual design calculations in Tables 6 to 9.

Since the section W10x49 is found to be unsatisfactory in cycle 1, a second cycle to test a heavier section (W10x66) is desired. This can be accomplished without termination of the program run. A value of 1 for the variable INDIC is input from the terminal (Fig. 8.15), indicating cycle 2 is required.

The externally input data items for this cycle are then input next via the terminal using the free-format type of input (Fig. 8.15). The amount of cycle 2 data is considerably less than that for cycle 1 since it only consists of data items whose values have been changed or new data items which the user wishes to add. For example, data number 59 (flange width) changed from 5.0



inch to 5.06 inch since the section has changed to a W10x66. Cycle 2 data items are reproduced on Fig. 8.22 for echo checking.

Once the program receives the data for this cycle, the checking procedure which follows is identical to that for cycle 1. The design message and data values shown in Figs. 8.26 and 8.27 at the end of this cycle indicate the section is satisfactory for the imposed loading. From Fig. 8.27, the values of equations (i), (ii) and (iii) are found to be 0.6999, 0.6984, and 0.8749 respectively. The correctness of these values is verified by the design calculations in Tables 6 to 12. The value of 0.87 for equation (iii) indicates the section is satisfactory but not overdesigned.





DESIGN CALCULATION FOR EXAMPLE 1  
AXIAL COMPRESSION MEMBER

SHEET 1 OF 2

LOADING

$$P_D = 300 \text{ kips} \quad \gamma = 0.9 \quad \alpha_T = 1.25$$

$$P_L = 85 \text{ kips} \quad \alpha_D = 1.25 \quad \psi = 1.0$$

$$P_Q = 0 \quad \alpha_L = 1.5$$

$$P_T = 0 \quad \alpha_R = 1.5$$

FACTORED COMPRESSIVE FORCE

$$\begin{aligned} P_f &= \gamma [\alpha_D P_D + \psi (\alpha_L P_L + \alpha_R P_Q + \alpha_T P_T)] && \text{(CLAUSE 7.2.2)} \\ &= 0.9 [1.25 \times 300 + 1.0 (1.5 \times 85)] \\ &= 452.25 \text{ kips} \end{aligned}$$

COLUMN SECTION 2 - L8x6x5/8  
G40.12 - 44W

$$\begin{aligned} L &= 12 \text{ ft.} & r_x &= 2.54 \text{ in} & S_x &= 197 \text{ in}^3 \\ A_g &= 16.7 \text{ in}^2 & r_y &= 2.42 \text{ in} \end{aligned}$$

CHECK CLASS OF SECTION

$$b/t = 8/63 = 12.7 \quad \therefore \text{CLASS} = 3 \quad \text{(CLAUSE 11)}$$

SWAY FORCES RESISTED BY SHEAR WALL

$$\therefore K_x = K_y = 1.0 \quad \text{(CLAUSE 9.3.2)}$$

$$\frac{K_x L_x}{r_x} = \frac{1.0 \times 144}{2.54} = 56.69$$

$$\frac{K_y L_y}{r_y} = \frac{1.0 \times 144}{2.42} = 59.5 \text{ (this Governs)}$$

TABLE 1 DESIGN CALCULATION FOR EXAMPLE 1



DESIGN CALCULATION FOR EXAMPLE 1  
AXIAL COMPRESSION MEMBER

SHEET 2 OF 2

$$\lambda = \frac{KL}{r} \sqrt{\frac{F_y}{\pi^2 E}} = 59.5 \sqrt{\frac{44}{\pi^2 \times 29000}} = 0.7377 \quad (\text{CLAUSE 13.3.1})$$

$$\begin{aligned} P_{rc} &= \phi A_g F_y (1.035 - 0.201 \lambda - 0.224 \lambda^2) \\ &= 0.9 \times 16.7 \times 44 (1.035 - 0.201 \times 0.7377 - 0.224 \times 0.7377^2) \\ &= 505.79 \text{ Kips} \end{aligned}$$

$$\begin{aligned} R1 &= P_f / P_{rc} \\ &= 452.25 / 505.79 \\ &= 0.8941 < (1.0) \end{aligned}$$

$\therefore$  STRENGTH CRITERION SATISFIED  
SECTION SATISFACTORY

TABLE 2 DESIGN CALCULATION FOR EXAMPLE 1



DESIGN CALCULATION FOR EXAMPLE 2  
LATERALLY UNSUPPORTED BEAM

SHEET 1 OF 3

LOADINGS

$$\text{DEAD LOAD} = 1.2 \text{ kips/ft.}$$

$$\text{LIVE LOAD} = 0.6 \text{ kips/ft.}$$

$$M_{xD} = \frac{1.2 \times 24^2 \times 12}{8} = 1036.8 \text{ in-kip}$$

$$M_{xL} = \frac{0.6 \times 24^2 \times 12}{8} = 518.4 \text{ in-kip}$$

$$M_{xR} = 0, \quad M_{xT} = 0$$

LOAD FACTORS

$$\gamma = 0.9, \quad \alpha_D = 1.25, \quad \alpha_L = 1.5, \quad \alpha_R = 1.5, \quad \alpha_T = 1.25, \quad \psi = 1.0$$

$$\begin{aligned} M_{fx} &= \gamma [\alpha_D M_{xD} + \psi (\alpha_L M_{xL} + \alpha_R M_{xR} + \alpha_T M_{xT})] \\ &= 0.9 [1.25 \times 1036.8 + 1.0 (1.5 \times 518.4)] \\ &= 1866.24 \text{ in-kip} \end{aligned}$$

CHECK W12x50 SECTION, 640.12-44W STEEL

SECTION PROPERTIES

$$F_y = 44 \text{ ksi}, \quad E = 29000 \text{ ksi}$$

$$\begin{aligned} d &= 12.19 \text{ in} & w &= 0.371 \text{ in} & I_y &= 56.4 \text{ in}^4 \\ b &= 4.04 \text{ in} & Z_x &= 72.5 \text{ in}^3 & A_f &= b \times t = 5.18 \text{ in}^2 \\ t &= 0.641 \text{ in} & S_x &= 64.8 \text{ in}^3 & A_e &= A_f + \frac{1}{6} w (d - 2t) \\ & & & & &= 5.85 \text{ in}^2 \end{aligned}$$

$$I_e = I_y / 2 = 56.4 / 2 = 28.2 \text{ in}^4 \quad (\text{NEGLECTING WEB CONTRIBUTION})$$

$$r_e = \sqrt{\frac{I_e}{A_e}} = \sqrt{\frac{28.2}{5.85}} = 2.2 \text{ in}$$

TABLE 3 DESIGN CALCULATION FOR EXAMPLE 2



DESIGN CALCULATION FOR EXAMPLE 2  
LATERALLY UNSUPPORTED BEAM

SHEET 2 OF 3

CHECK SECTION CLASS

$$b/t = 4.04 / 0.641 = 6.30 < 54 / \sqrt{F_y} \quad \therefore \text{FLANGE CLASS} = 1 \quad (\text{CLAUSE 11})$$

$$h/w = (d - 2t) / w = \frac{12.19 - 2 \times 0.641}{0.371} = 29.4 < 420 / \sqrt{F_y} \quad \therefore \text{Web CLASS} = 1 \quad (\text{CLAUSE 11})$$

$$\therefore \text{CLASS OF SECTION} = 1$$

CALCULATE  $M_u$

$$\sigma_1 = \frac{20000}{L d / A_f} = \frac{20000 \times 5.18}{24 \times 12 \times 12.19} = 29.51$$

$$\sigma_2 = \frac{250000}{(L / r_g)^2} = \frac{250000 \times 2.2^2}{(24 \times 12)^2} = 14.59$$

$$\omega_x = 1.0 \quad (\text{CLAUSE 13.8.3 (a)})$$

$$M_u = \frac{S_x}{\omega_x} \sqrt{\sigma_1^2 + \sigma_2^2} = \frac{64.8}{1.0} \sqrt{29.51^2 + 14.59^2} = 2133.2 \text{ in-Kip} \quad (\text{CLAUSE 13.6.1})$$

$$M_p = Z_x F_y = 7.25 \times 44 = 3190 \text{ in-Kip}$$

$$\therefore M_u > 2/3 M_p$$

$$M_{rx} = 1.15 \phi M_p \left( \frac{1 - 0.28 M_p}{M_u} \right) \quad (\text{CLAUSE 13.6.1 (a)})$$

$$= 1.15 \times 0.9 \times 3190 \left( 1 - \frac{0.28 \times 3190}{2133.2} \right) = 1919.2 \text{ in-Kip}$$

$$\therefore R2 = M_{fx} / M_{rx} = 1866.24 / 1919.2 = 0.97 < (1.0)$$

$$\therefore \text{STRENGTH CRITERION SATISFIED}$$

TABLE 4 DESIGN CALCULATION FOR EXAMPLE 2





DESIGN CALCULATION FOR EXAMPLE 2  
LATERALLY UNSUPPORTED BEAM

SHEET 3 OF 3

CHECK SHEAR

$$V_D = \frac{1.2 \times 24}{2} = 14.4 \text{ Kips}$$

$$V_L = \frac{0.6 \times 24}{2} = 7.2 \text{ Kips}$$

$$V_e = 0, \quad V_T = 0$$

$$\begin{aligned} V_f &= 0.9 [1.25 \times 14.4 + 1.0 (1.5 \times 7.2)] \\ &= 25.92 \text{ Kips} \end{aligned}$$

$$h/w = 29.4 < 167 \sqrt{\frac{K_v}{F_y}} = 58.18 \quad (\text{CLAUSE 13.4.1})$$

( $K_v = 5.34$  for unstiffened webs)

$$\therefore F_s = 0.66 F_y$$

$$\begin{aligned} V_r &= \phi A_w F_s = 0.9 \times 4.05 \times 0.66 \times 44 \\ &= 105.85 \text{ Kips} \end{aligned}$$

$$\frac{V_f}{V_r} = \frac{25.92}{105.85} = 0.24 (< 1.0)$$

$\therefore$

SHEAR CRITERION SATISFIED

W12x50

SECTION SATISFACTORY



# DESIGN CALCULATION FOR EXAMPLE 3 AXIAL COMPRESSION AND BENDING

SHEET 1 OF 7

## LOADING

$$\begin{array}{lll}
 P_D = 90 \text{ kips} & M_{xD} = 0 & M_{yD} = 0 \\
 P_L = 21 \text{ kips} & M_{xL} = 1067 \text{ in-kip} & M_{yL} = 297 \text{ in-kip} \\
 P_Q = 0 & M_{xQ} = 0 & M_{yQ} = 0 \\
 P_T = 0 & M_{xT} = 0 & M_{yT} = 0
 \end{array}$$

## LOAD FACTORS

$$\gamma = 0.9, \alpha_D = 1.25, \alpha_L = 1.5, \alpha_Q = 1.5, \alpha_T = 1.25, \psi = 1.0$$

$$P_f = 0.9[1.25 \times 90 + 1.0(1.5 \times 21)] = 129.6 \text{ kips}$$

$$M_{fx} = 0.9 \times 1.5 \times 1067 = 1440.45 \text{ in-kips}$$

$$M_{fy} = 0.9 \times 1.5 \times 297 = 400.95 \text{ in-kips}$$

## CYCLE NUMBER 1

CHECK W10X49 SECTION G40.12 - 44 W STEEL

## SECTION PROPERTIES

$$\begin{array}{lll}
 d = 10.0 \text{ in} & Z_x = 60.3 \text{ in}^3 & A_f = 5.58 \text{ in}^2 \\
 b = 5.0 \text{ in} & S_x = 54.6 \text{ in}^3 & I_e = I_y/2 = 46.5 \text{ in}^4 \\
 t = 0.558 \text{ in} & A_g = 14.4 \text{ in}^2 & A_e = A_f + 1/6 w(d - 2t) \\
 w = 0.34 \text{ in} & r_x = 4.35 \text{ in} & = 6.08 \text{ in}^2 \\
 h = 8.88 \text{ in} & r_y = 2.54 \text{ in} & r_e = \sqrt{I_e/A_e} = 2.77 \text{ in} \\
 F_y = 44 \text{ ksi}, E = 29000 \text{ ksi} & & S_y = 18.6 \text{ in}^3 \\
 & & Z_y = 28.2 \text{ in}^3
 \end{array}$$

## CHECK SECTION CLASS

$$P_f/C_y = 129.6 / 14.4 \times 44 = 0.20 > 0.15$$

$$b/t = 5.0 / 0.558 = 8.96 \left( > \frac{54}{\sqrt{F_y}}, < \frac{64}{\sqrt{F_y}} \right) \therefore \text{FLANGE CLASS} = 2$$

$$h/w = 8.88 / 0.34 = 26.12 \left( < \frac{420}{\sqrt{F_y}} (1 - 1.4 P_f/C_y) \right) \therefore \text{WEB CLASS} = 1 \quad \therefore \text{SECTION CLASS} = 2$$



DESIGN CALCULATION FOR EXAMPLE 3  
AXIAL COMPRESSION AND BENDING

SHEET 2 OF 7

CALCULATION OF RESISTANT FORCES

SWAY FORCES IN BOTH DIRECTIONS RESISTED BY BRACING SYSTEMS

$$\therefore K_x = K_y = 1.0 \quad (\text{CLAUSE 9.3})$$

$$L_x = L_y = 114 \text{ in}$$

$$\begin{aligned} U_{KL R} &= \text{MAX} \left( \frac{L_x L_x}{r_x}, \frac{K_y L_y}{r_y} \right) \\ &= \text{MAX} \left( \frac{1.0 \times 114}{4.35}, \frac{1.0 \times 114}{2.54} \right) = 44.88 \end{aligned}$$

$$\begin{aligned} \therefore \lambda_1 &= U_{KL R} \sqrt{\frac{F_y}{\pi^2 E}} \quad (\text{CLAUSE 13.3.1}) \\ &= 44.88 \sqrt{\frac{44}{\pi^2 \times 29000}} = 0.557 \end{aligned}$$

$$\begin{aligned} P_{rc1} &= \phi A_g F_y (1.035 - 0.201 \lambda_1 - 0.224 \lambda_1^2) \quad (\text{CLAUSE 13.3.1}) \\ &= 0.9 \times 14.4 \times 44 (1.035 - 0.201 \times 0.557 - 0.224 \times 0.557^2) \\ &= 486.78 \text{ kips} \end{aligned}$$

$$P_{rc2} = \phi A_g F_y = 0.9 \times 14.4 \times 44 = 570.24 \text{ kips} \quad (\text{CLAUSE 13.8.2(ii)})$$

$$\begin{aligned} M_{p(\text{reduced})} &= 1.18 M_p (1 - P_f / C_y) \\ &= 1.18 \times 44 \times 60.3 \left( 1 - \frac{129.6}{14.4 \times 44} \right) \\ &= 2490.39 \text{ in- kip} \end{aligned}$$

$$\begin{aligned} M_{rx1} &= \phi M_{p(\text{reduced})} \\ &= 0.9 \times 2490.39 \\ &= 2241.35 \text{ in- kip} \end{aligned}$$

$$M_{vy1} = 1116.7 \text{ in- kip} \quad (\text{CLAUSE 13.6.1})$$



DESIGN CALCULATION FOR EXAMPLE 3  
AXIAL COMPRESSION AND BENDING

SHEET 3 OF 7

$$M_u = \frac{S_x}{\omega_x} \sqrt{\sigma_1^2 + \sigma_2^2} \quad (\text{CLAUSE 13.6.1})$$

$$\sigma_1 = \frac{20000}{L_d/A_f} = \frac{20000 \times 5.58}{114 \times 10} = 97.89$$

$$\sigma_2 = \frac{250000}{(L/r_x)^2} = \frac{250000 \times 2.77^2}{114^2} = 147.6$$

$$\omega_x = 1.0$$

$$\omega_y = 0.85$$

(CLAUSE 13.8.3.(a) (ii))

$$M_u = \frac{54.6}{1.0} \sqrt{97.89^2 + 147.6^2} = 9670.3 \text{ in-kip}$$

$$M_u > 2/3 M_p (\text{reduced})$$

$$\begin{aligned} \therefore M_{rx2} &= 1.15 \phi M_p(\text{reduced}) \left( 1 - \frac{0.28 M_p(\text{reduced})}{M_u} \right) \quad (\text{CLAUSE 13.6.1}) \\ &= 1.15 \times 0.9 \times 2490.39 \left( 1 - \frac{0.28 \times 2490.39}{9670.3} \right) \\ &= 2391.69 \text{ in-kip} > \phi M_p(\text{reduced}) \end{aligned}$$

$$\begin{aligned} \therefore M_{rx2} &= \phi M_p(\text{reduced}) = 0.9 \times 2490.39 \\ &= 2241.35 \text{ in-kip} \end{aligned}$$

$$C_{ex} = \frac{286000 A_g}{(K_x L_x / r_x)^2} = 5995.07$$

$$C_{ey} = \frac{286000 A_g}{(K_y L_y / r_y)^2} = 2044.67$$

TABLE 8 DESIGN CALCULATION FOR EXAMPLE 3





DESIGN CALCULATION FOR EXAMPLE 3  
AXIAL COMPRESSION AND BENDING

SHEET 4 OF 7

$$\frac{M_{fx}}{M_{rx1}} + \frac{M_{fy}}{M_{ry1}} = \frac{1440.45}{2241.35} + \frac{400.95}{1116.7} \quad (\text{CLAUSE 13.8.2 (i)})$$

$$= 0.6427 + 0.359 = 1.0017 (>1.0)$$

STRENGTH CRITERION NOT SATISFIED

$$\frac{P_f}{P_{rx1}} + \frac{0.85 M_{fx}}{M_{rx1}} + \frac{0.6 M_{fy}}{M_{ry1}} = \frac{129.6}{570.24} + \frac{0.85 \times 1440.45}{2241.35} + \frac{0.6 \times 400.95}{1116.7} \quad (\text{CLAUSE 13.8.2 (ii)})$$

$$= 0.2273 + 0.5463 + 0.2153 = 0.9888 (<1.0)$$

STRENGTH CRITERION SATISFIED

$$\frac{P_f}{P_{rx1}} + \frac{\omega_x M_{fx}}{M_{rx1} (1 - P_f / C_{ex})} + \frac{\omega_y M_{fy}}{M_{ry1} (1 - P_f / C_{ey})} = \frac{129.6}{486.78} + \frac{1.0 \times 1440.45}{2241.35 (1 - 129.6 / 5995.07)} + \frac{0.85 \times 400.95}{1116.7 (1 - 129.6 / 2044.67)}$$

$$= 0.27 + 0.6593 + 0.1623$$

$$= 1.0916 (>1.0)$$

STABILITY CRITERION NOT SATISFIED

W10x49 SECTION IS NOT SATISFACTORY

TABLE 9 DESIGN CALCULATION FOR EXAMPLE 3



DESIGN CALCULATION FOR EXAMPLE 3  
AXIAL COMPRESSION AND BENDING

SHEET 5 OF 7

CYCLE NUMBER 2

CHECK W10x66 G40.12 - 44W STEEL SECTION

SECTION PROPERTIES

$$\begin{array}{lll} d = 10.38 \text{ in} & h = 8.88 \text{ in} & r_x = 4.44 \text{ in} \\ b = 5.06 \text{ in} & S_x = 73.6 \text{ in}^3 & r_y = 2.58 \text{ in} \\ t = 0.748 \text{ in} & A_g = 19.4 \text{ in}^2 & A_f = 7.57 \text{ in}^2 \\ w = 0.457 \text{ in} & Z_x = 82.8 \text{ in}^3 & r_b = 2.8 \text{ in} \end{array}$$

CHECK SECTION CLASS

$$P_f / C_y = 129.6 / 853.6 = 0.1518 (> 0.15)$$

$$b/t = 5.06 / 0.748 = 6.76 (< 54 / \sqrt{F_y} = 8.14) \quad \text{FLANGE CLASS} = 1$$

$$h/w = 8.88 / 0.457 = 19.43 (< \frac{420}{\sqrt{F_y}} (1.4 P_f / C_y)) \quad \text{WEB CLASS} = 1$$

$\therefore$  CLASS OF SECTION = 1

CALCULATION OF RESISTANT FORCES

$$\begin{aligned} U_{KL R} &= \max \left( \frac{L_x L_x}{r_x}, \frac{L_y L_y}{r_y} \right) \\ &= \max (25.68, 44.19) \\ &= 44.19 \end{aligned}$$

$$\lambda_1 = 44.19 \sqrt{\frac{44}{\pi^2 \times 29000}} = 0.548$$

$$\begin{aligned} P_{rc1} &= \phi A_g F_y (1.035 - 0.201 \lambda_1 - 0.224 \lambda_1^2) \\ &= 0.9 \times 19.4 \times 44 (1.035 - 0.201 \times 0.548 - 0.224 \times 0.548^2) \\ &= 658.83 \text{ kips} \end{aligned}$$

$$\begin{aligned} P_{rc2} &= \phi A_g F_y = 0.9 \times 19.4 \times 44 \\ &= 768.24 \text{ kips} \end{aligned}$$



DESIGN CALCULATION FOR EXAMPLE 3  
AXIAL COMPRESSION AND BENDING

SHEET 6 OF 7

$$\begin{aligned} M_{p(\text{reduced})} &= 1.18 \times M_p (1 - P_f / C_y) \\ &= 1.18 \times 3643.2 (1 - 0.1518) \\ &= 3646.27 \text{ in-kip} \end{aligned}$$

$$M_{rx1} = \phi M_{p(\text{reduced})} = 0.9 \times 3646.27 = 3281.64 \text{ in-kip}$$

$$\sigma_1 = 20000 / (L_d / A_f) = \frac{20000 \times 7.57}{114 \times 10.38} = 127.95$$

$$\sigma_2 = 250000 / (L / r_e)^2 = \frac{250000 \times 2.8^2}{114^2} = 150.82$$

$$M_u = \frac{S_x}{W_x} \sqrt{\sigma_1^2 + \sigma_2^2} = \frac{73.6}{1.0} \sqrt{127.95^2 + 150.82^2} = 14556.3 \text{ in-kip}$$

$$M_u > 2/3 M_{p(\text{reduced})}$$

$$\begin{aligned} \therefore M_{rx2} &= 1.15 \phi M_{p(\text{reduced})} \left( 1 - \frac{0.28 M_{p(\text{reduced})}}{M_u} \right) \\ &= 1.15 \times 0.9 \times 3646.27 \left[ 1 - \frac{0.28 \times 3646.27}{14556.3} \right] \\ &= 3509.19 \text{ in-kip} > (\phi M_{p(\text{reduced})}) \end{aligned}$$

$$\therefore M_{rx2} = \phi M_{p(\text{reduced})} = 3281.64 \text{ in-kip}$$

$$C_{ex} = \frac{286000 \times 19.4}{25.68^2} = 8416.3$$

$$C_{ey} = \frac{286000 \times 19.4}{44.19^2} = 2841.3$$



DESIGN CALCULATION FOR EXAMPLE 3  
AXIAL COMPRESSION AND BENDING

SHEET 7 OF 7

$$\frac{M_{fx}}{M_{rx1}} + \frac{M_{fy}}{M_{ry1}} = \frac{1440.45}{3281.64} + \frac{400.95}{1536.48} = 0.4398 + 0.26$$

$$= 0.6999 \quad (< 1.0)$$

STRENGTH CRITERION SATISFIED

$$\frac{P_f}{P_{rc1}} + \frac{0.85 M_{fx}}{M_{rx1}} + \frac{0.6 M_{fy}}{M_{ry1}} = \frac{129.6}{768.24} + \frac{0.85 \times 1440.45}{3281.64} + \frac{0.6 \times 400.95}{1536.48}$$

$$= 0.1687 + 0.3731 + 0.1566$$

$$= 0.6984 \quad (< 1.0)$$

STRENGTH CRITERION SATISFIED

$$\frac{P_f}{P_{rc1}} + \frac{\omega_x M_{fx}}{M_{rx2}(1 - P_f/C_{ex})} + \frac{\omega_y M_{fy}}{M_{ry2}(1 - P_f/C_{ey})}$$

$$= \frac{129.6}{658.83} + \frac{1.0 \times 1440.45}{3281.64(1 - 129.6/84163)} + \frac{0.85 \times 400.95}{1536.48(1 - 129.6/2841.3)}$$

$$= 0.1967 + 0.4458 + 0.2324$$

$$= 0.8749 \quad (< 1.0)$$

STABILITY CRITERION SATISFIED

SECTION W10x66 SATISFACTORY

TABLE 12 DESIGN CALCULATION FOR EXAMPLE 3





## CHAPTER IX

### SUMMARY AND CONCLUSIONS

Currently in engineering, the development of computer programs in analysis is much more advanced than that in design and decision processing. The computer will be an even more valuable tool to engineers if it can be used for implementing decisions to a greater extent.

A large part of the design process in structural engineering is devoted to satisfying specific requirements of the codes or specifications of different authorities. This procedure largely involves checking of logical conditions, arising from numerical calculations.

The method of tabular decision logic used in this thesis is found to be a suitable technique for formulating, displaying, and documenting the decision making procedures required by codes and specifications. Since this technique presents decision making procedures in a logical fashion, it is also useful for implementing these procedures on the computer.

Most computer-aided structural design programs in use at the present time, such as the CISC - Column Selection Program (3), and the CISC - Floor System Selection Program (4), have the code requirements built into the analysis and member selection subroutines.



Since code requirements are often subjected to revisions, revising and updating these programs becomes a tedious task.

The decision table technique used herein to compile sections of the new CSA S16.1 - Steel Structures for Buildings - Limit States Design Standard renders the code requirement checking operation into an individual module entirely independent from the analysis and member selection procedure. In this way, any further code revisions can easily be handled by changing the relevant decision tables and their associated condition and action subroutines (Sect. 5.4). This procedure is schematically illustrated in Fig. 9.1.

In Chapter VIII, a number of example problems are checked against CSA S16.1 using the decision tables compiled in this thesis. The validity of the tables is checked by manual calculations. The design decisions incorporated into the decision tables are completely objective and according to the requirements of the Standard.

Conditional execution has been used extensively in the decision table arrangement. The main reason for this approach is that in conditional execution, not every data item required by a particular decision table need be available before the execution can commence. This is in contrast to the direct execution approach which requires the presence of all the data items appearing in the table



regardless of whether they are actually used. In addition, often due to the presence of immaterial conditions in the condition entries of a table, certain conditions may not have to be tested at all in order to locate the governing rule. The precomputation of such data elements is, therefore, wasteful. Hence, this approach is inefficient and results in a large number of redundant data items.

An interactive mode procedure has been developed herein which enables the user to execute program runs and input data items from an on-line terminal remote from the physical location of the computer. This mode also facilitates the input of second or subsequent cycles of data immediately after the result of the previous cycle is known.

A pilot scheme for the recursive execution of tables has been introduced. This leads to the possibility that a considerable saving in the total number of data elements and decision tables may be achieved.

There are a number of areas in which improvements or further developments of the work described in this thesis can be made. Some of these are:

- (a) To complete the documentation of CSA S16.1, in decision table format, so that it can be incorporated into analysis and design programs.
- (b) To associate the code requirement checking processor with a module which sizes and selects



members according to design criteria so that the design process can be more fully automated.

- (c) The recursive execution of tables within a cycle increases the complexity of the ingredience-dependence relationship of data elements. A more efficient dynamic concept of ingredience and dependence may be desirable.
- (d) The input and output may be improved to be more user orientated. If the output and input of data elements can be referenced by name as well as subscript, it will undoubtedly be helpful to the user when interpreting output results. It would also be practical if the user could select whichever data element he desires to be output on the terminal for checking and reviewing.
- (e) In a completely integrated scheme, the checking program should have a provision which automatically returns control back to the member selection module if a section does not satisfy the code requirements or is overdesigned, as illustrated in Fig. 9.1. Furthermore, improvements can be made in the interactive mode to provide a flexible system which allows the designer to 'break in' to a cycle at any point he desires.





## FIGURES



CONDITION STUB	CONDITION ENTRIES	INGREDIENT LIST OF EACH CONDITION
ACTION STUB	ACTION ENTRIES	INGREDIENT LIST OF EACH ACTION

FIG. 2.1 BASIC COMPONENTS OF A DECISION TABLE

CONDITION 1	Y	N	N	Y
CONDITION 2	N	Y	N	Y
ACTION 1	Y			
ACTION 2		Y	Y	
ACTION 3				Y

FIG. 2.2 A COMPLETE DECISION TABLE



CONDITION 1	Y	N	Y
CONDITION 2	N	I	Y
ACTION 1	Y		
ACTION 2		Y	
ACTION 3			Y

FIG. 2.3 A REDUCED SIZE DECISION TABLE

Definition of data item	External input flag	Table number from which value of data can be obtained
----------------------------	------------------------	---

FIG. 2.4 DATA REQUIREMENTS OF A DECISION TABLE



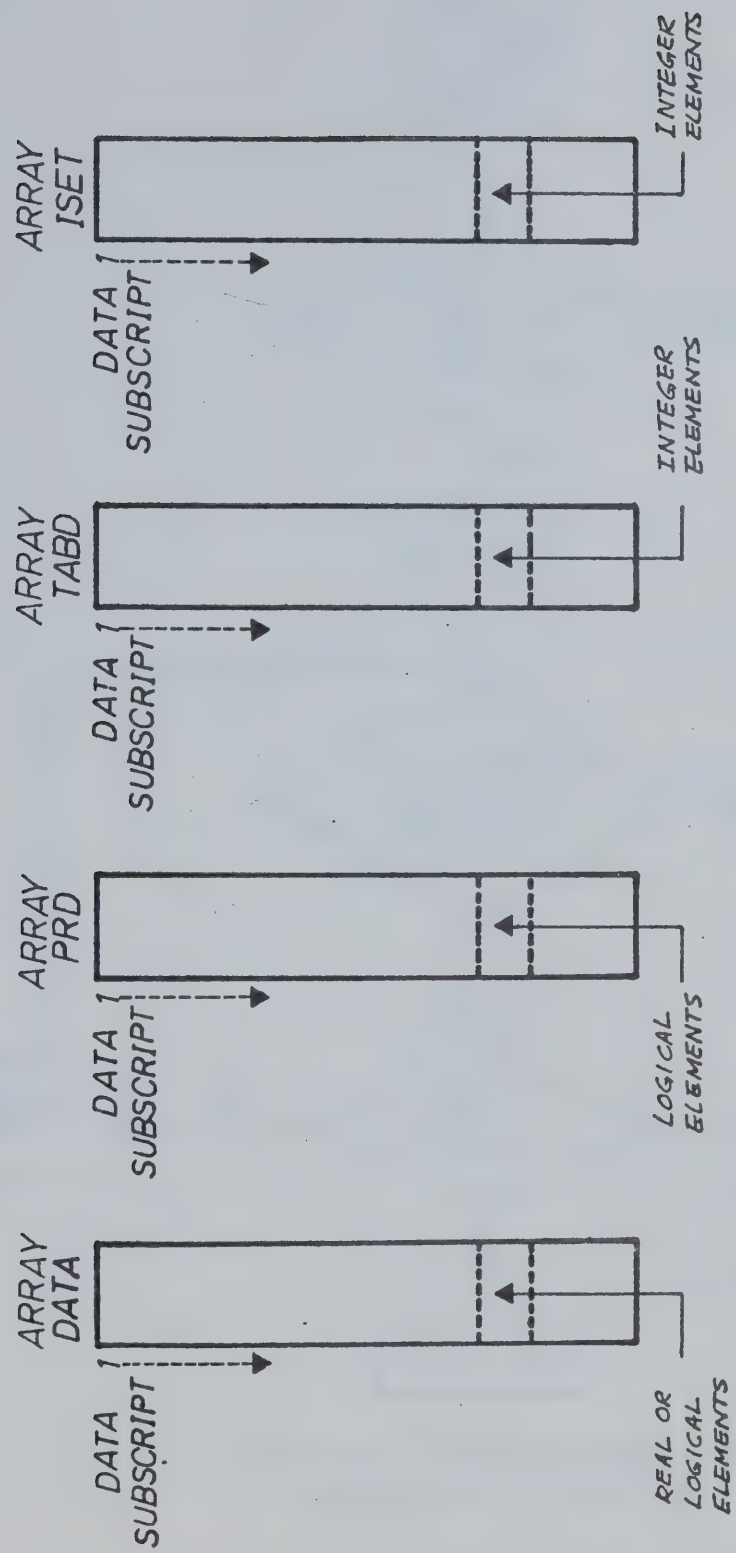


FIG. 3.1 GLOBAL ARRAYS





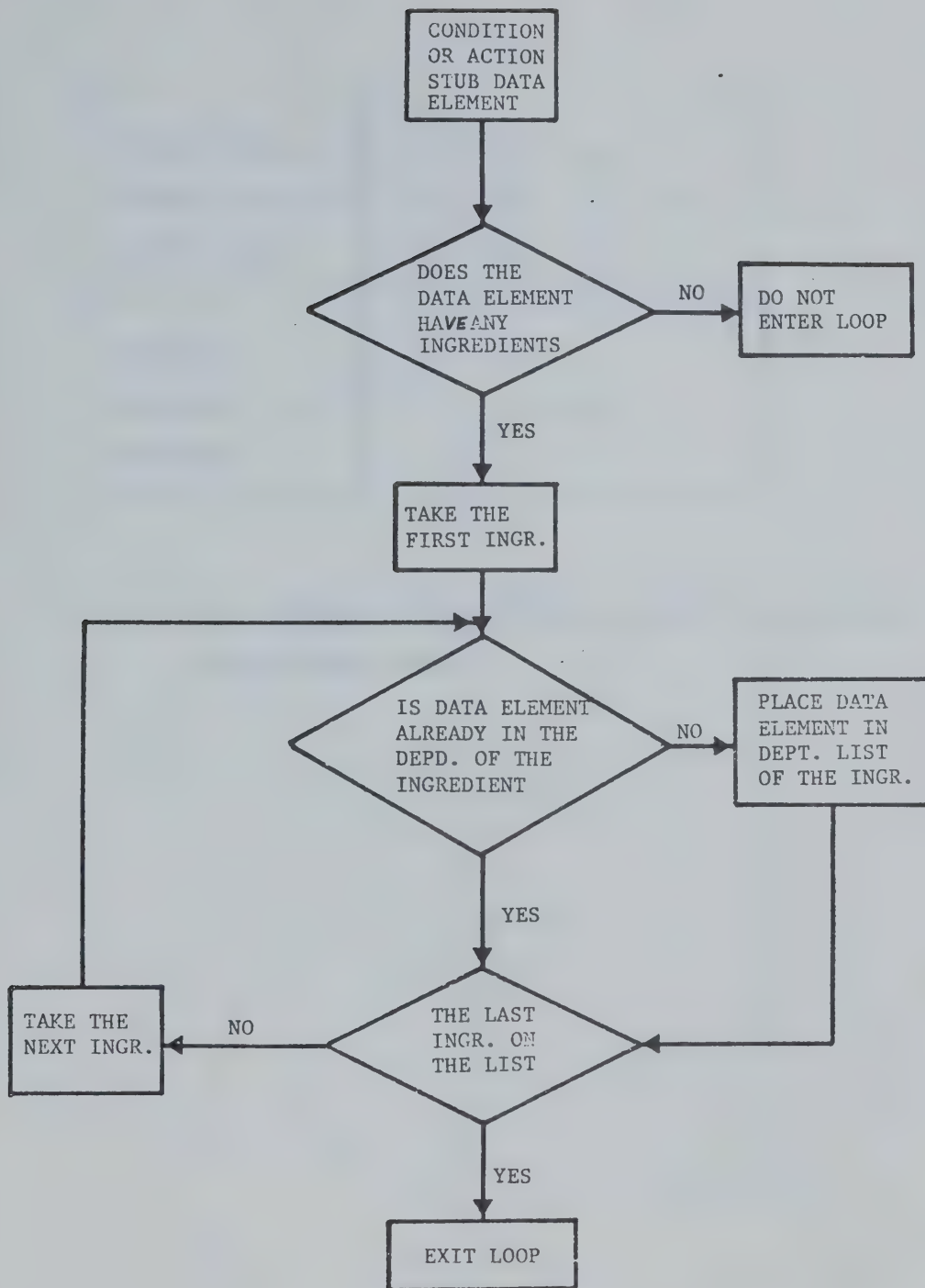


FIG. 3.2 GENERATING DEPENDENTS FROM INGREDIENTS OF A CONDITION OR AN ACTION



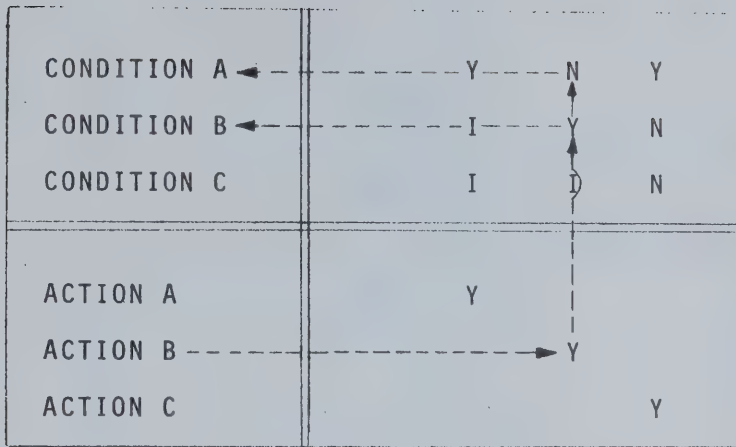


FIG 3.3 ILLUSTRATION OF DEPENDENCE CONCEPT BY THE LOGIC OF DECISION TABLES



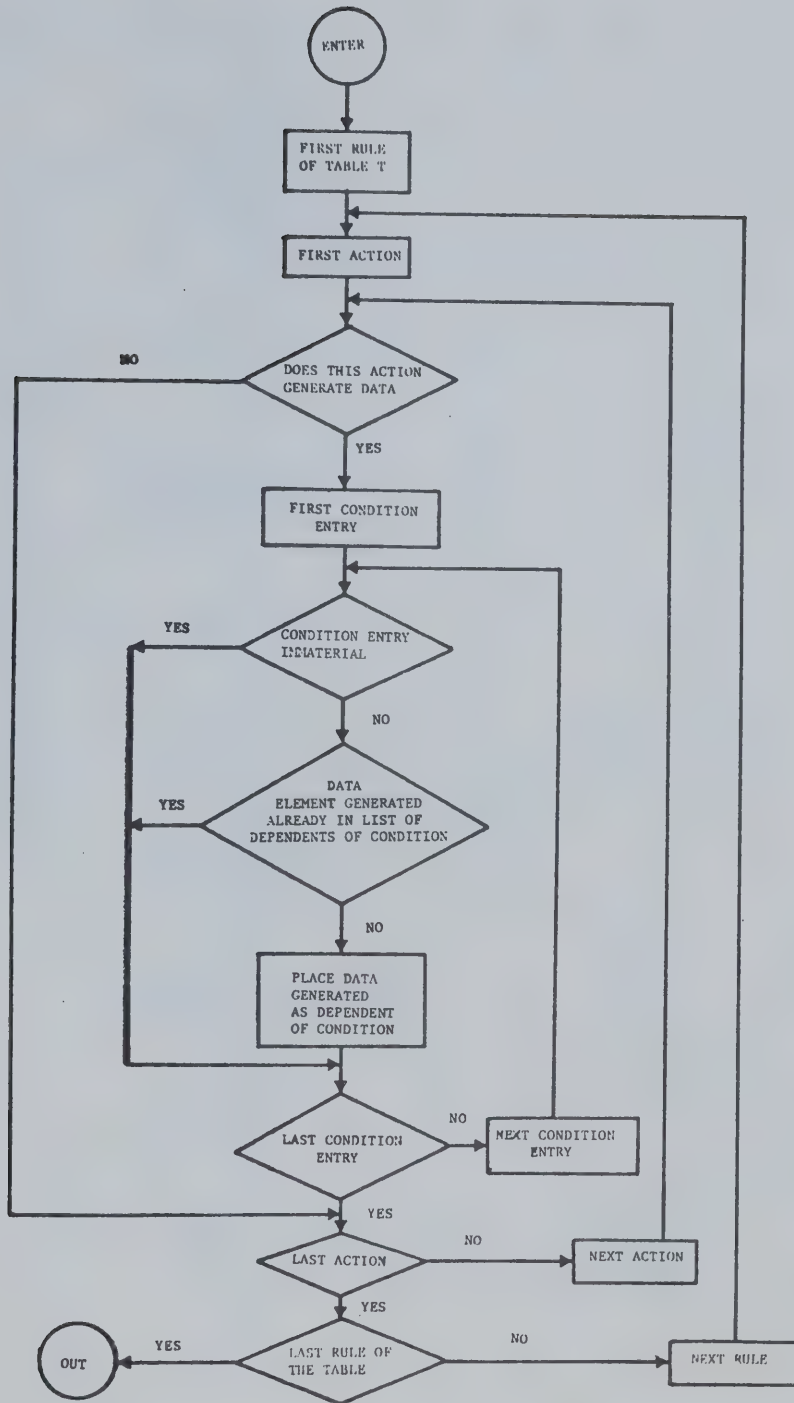


FIG. 3.4 FLOW CHART FOR GENERATING DEPENDENTS BY THE LOGIC OF DECISION TABLES



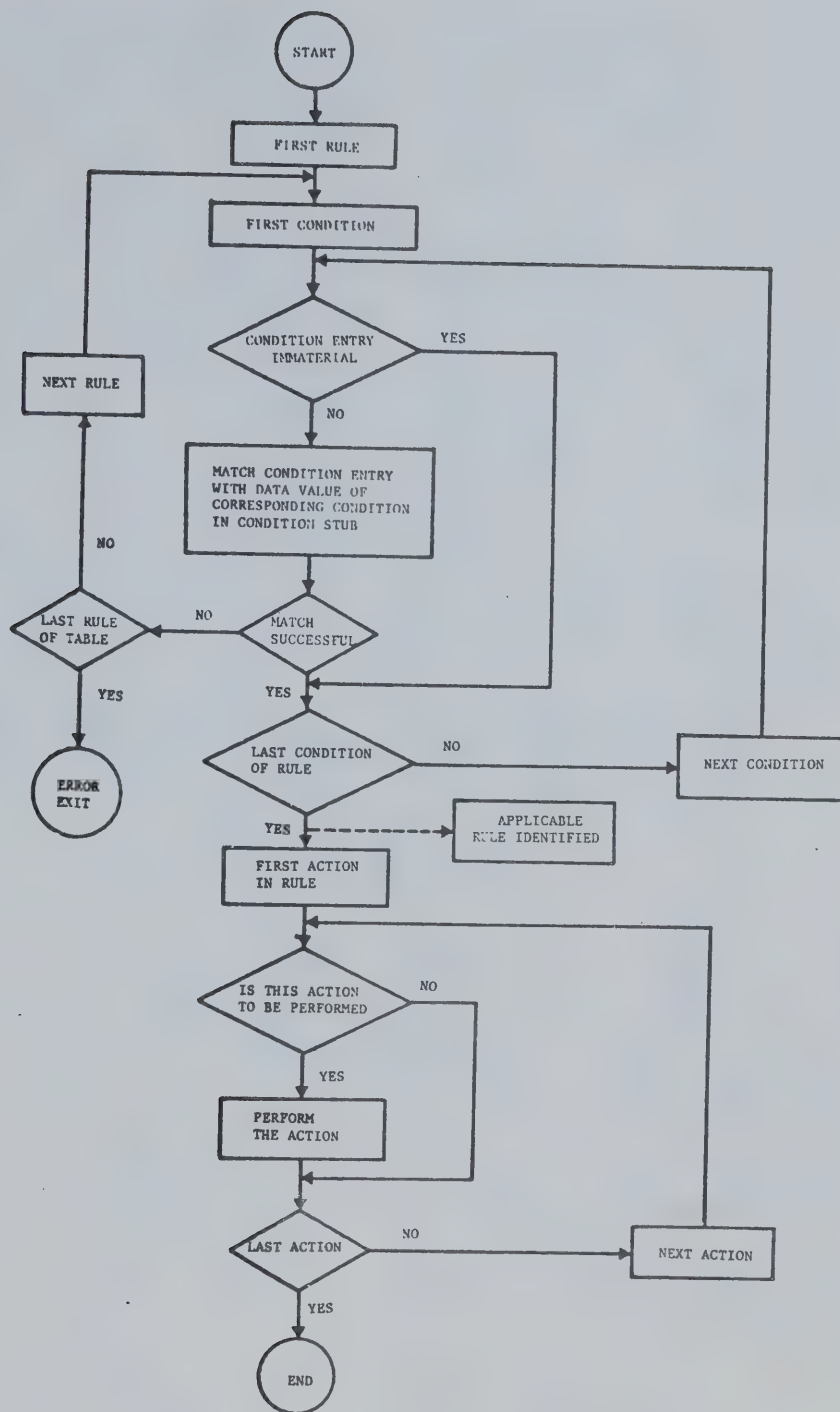


FIG. 4.1 PROCESSING OF A DECISION TABLE





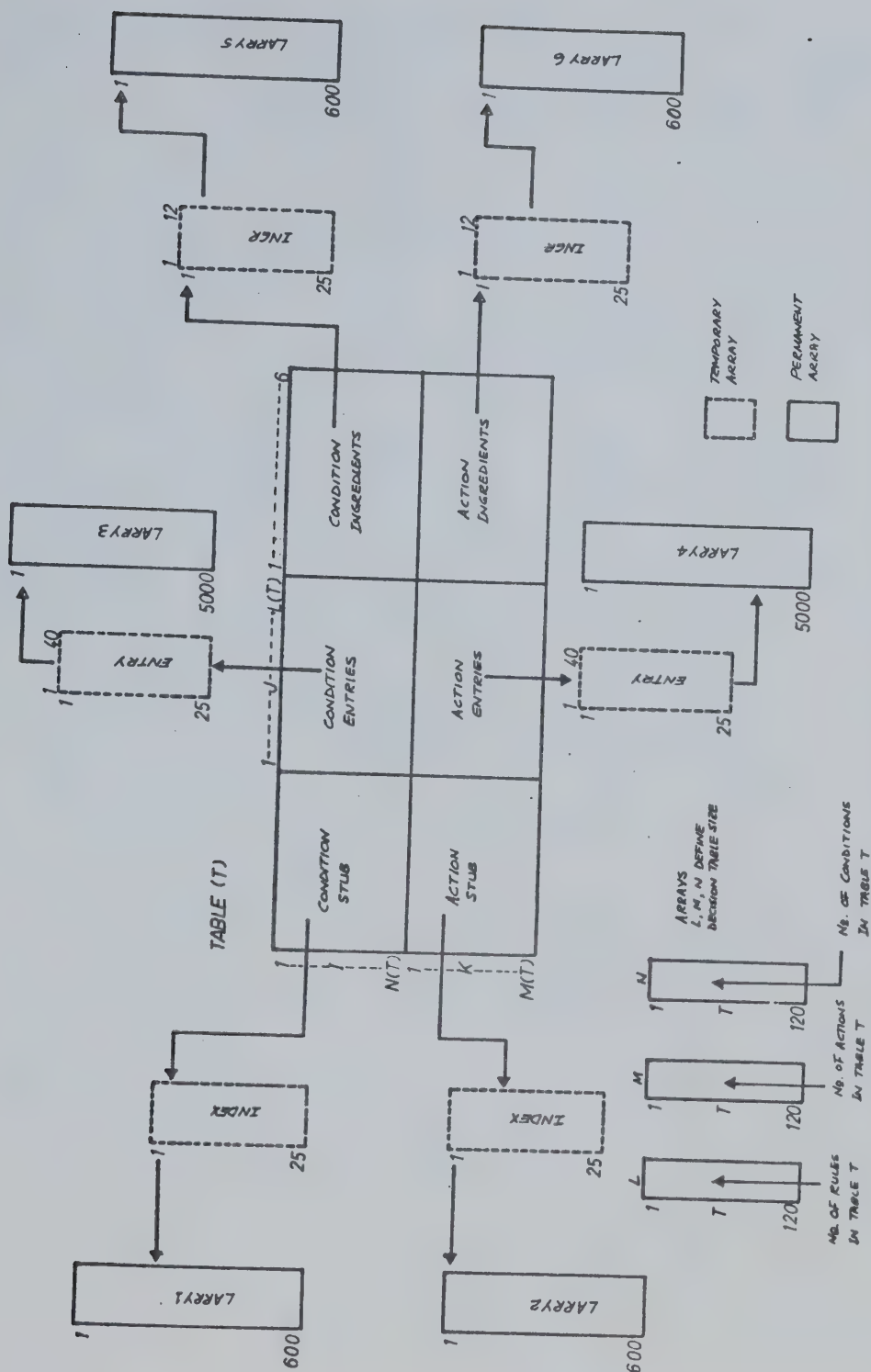


FIG. 5.1 PERMANENT AND TEMPORARY DECISION TABLE STORAGE ARRAYS



CONTENT	TEMPORARY ARRAY	PERMANENT ARRAY	POINTER ARRAY
Condition Stubs	INDEX	LARRY1	IBASE(T,1)
Action Stubs	INDEX	LARRY2	IBASE(T,2)
Condition Entries	ENTRY	LARRY3	IBASE(T,3)
Action Entries	ENTRY	LARRY4	IBASE(T,4)
Condition Ingredients	INGR	LARRY5	IPNTRC
Action Ingredients	INGR	LARRY6	IPNTRA
Mutually Exclusive Sets	TMPSET	MEXSET	MARCA
Dependents	IDEPND	ICLEAR	IARROW

FIG. 5.2 PERMANENT, TEMPORARY ARRAYS  
AND THEIR POINTER ARRAYS



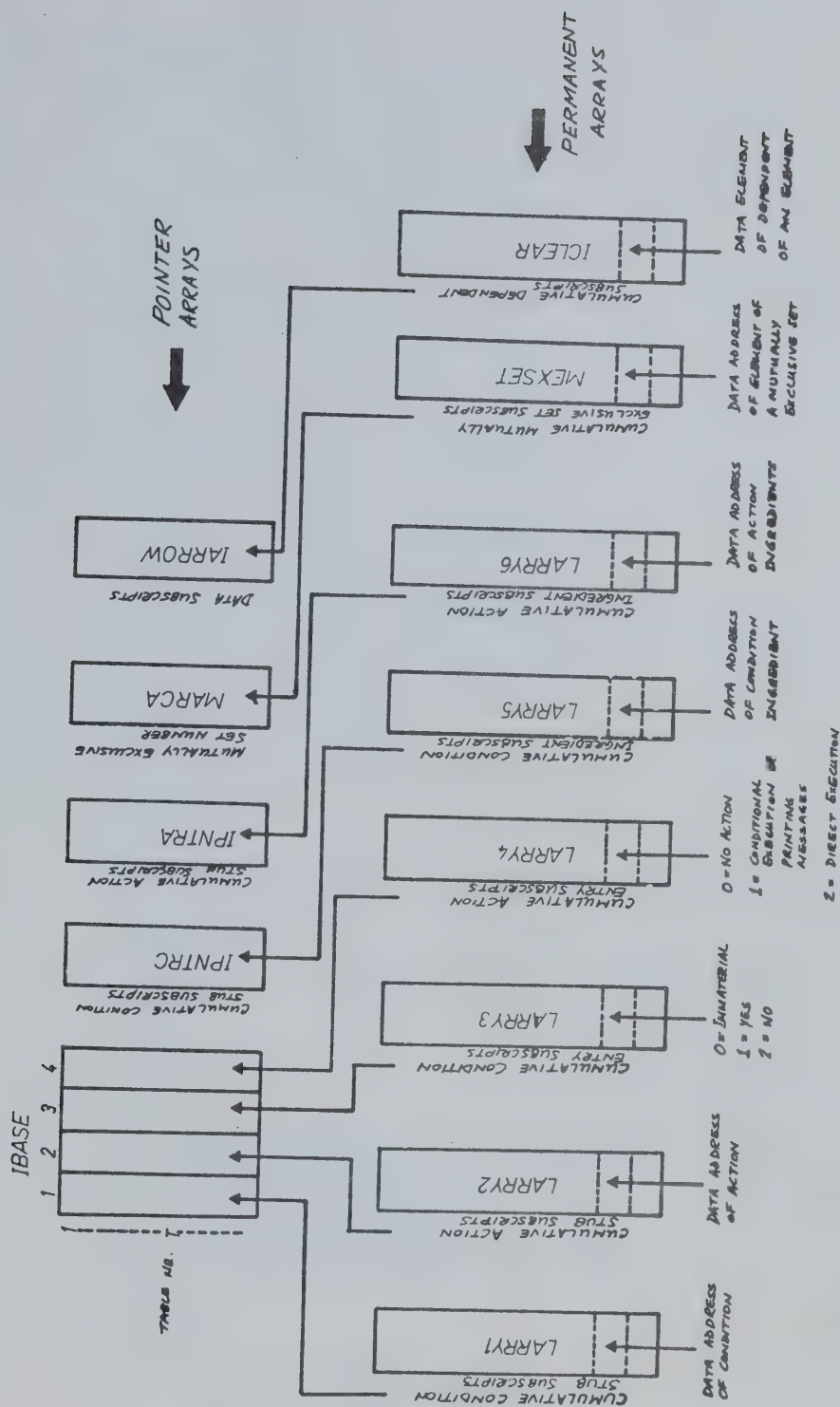


FIG. 5.3 OVERALL ARRAY STRUCTURE AND INTERRELATIONSHIP



	44	4	10	
110	1221222222			
111	2122122222			
112	2212212222			
113	222221111			
99	222221222			
100	222222122			
101	222222212			
102	1111112221			
274	111222	63	64	75
272	222111	63	64	75
134	111111	197	80	76
134	111	197	80	76
		63	64	47
				48C
151	111111111	132	134	

FIG. 5.4 CODED INPUT FORM OF DECISION TABLE 13.5.A.1 (44)





```

SUBROUTINE CC44(I)
  IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
  COMMON/HICA/DATA(700),PRD(700)
  EQUIVALENCE (DATA(63),SH), (DATA(64),SW), (DATA(76),FY),
  * (DATA(272), $HW13), (DATA(273), $HW14), (DATA(75),PCR),
  * (DATA(274), $HW15)
  GO TO (9999,9999,9999,9999,9999,9999,9999,9999,90,100),I
90 IF(SH/SW.GT.12000.0/FY) GO TO 102
  IF(SH/SW.LE.690.0/SQRT(PCR)) GO TO 102
  $HW15=1.0
  GO TO 104
102 $HW15=0.0
104 PRD(274)=.TRUE.
  RETURN
100 IF(SH/SW.LE.690.0/SQRT(PCR)) GO TO 106
  $HW13=0.0
  GO TO 108
106 $HW13=1.0
108 PRD(272)=.TRUE.
  RETURN
9999 WRITE(6,110)
110 FORMAT(1X,'NO SUBROUTINE NECESSARY FOR THIS CONDITION')
  RETURN
  END
SUBROUTINE AA44(K)
  IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
  COMMON/HICA/DATA(700),PRD(700)
  COMMON/HINCE/ICycle
  EQUIVALENCE (DATA(134),PMRX), (DATA(197),PHAI), (DATA(80),S),
  * (DATA(75),PCR), (DATA(47),AW), (DATA(48),AF), (DATA(63),SH),
  * (DATA(64),SW), (DATA(151),R2), (DATA(132),PMFX), (DATA(76),FY)
  GO TO (10,20,30,40),K
10 PMRX=AMIN1(PHAI*S*PCR,PHAI*S*FY)
  PRD(134)=.TRUE.
  RETURN
20 PMRX=AMIN1(PHAI*S*FY,PHAI*S*PCR)*(1.0-0.0005*AW/AF*(SH/SW-690.0/
  *SQRT(PCR)))
  PRD(134)=.TRUE.
  RETURN
30 WRITE(6,100)
  WRITE(7,100)
100 FORMAT(1X,'***** PMRX=DATA(134), TO BE DETERMINED BY CLAUSE 12,',
  * ' CSA S136. *****')
  WRITE(6,102)
  WRITE(7,102)
102 FORMAT(1H1,2X,'*****CALL OUTPUT, THEN TERMINATE PROGRAM',
  * ' BECAUSE OF THE ABOVE MESSAGE FROM AA44*****')
  CALL OUTPUT(ICycle)
  STOP
40 R2=PMFX/PMRX
  PRD(151)=.TRUE.
  RETURN
  END
SUBROUTINE CC45(I)
  IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
  COMMON/HICA/DATA(700),PRD(700)
  EQUIVALENCE (DATA(59),SB), (DATA(58),ST), (DATA(191),SKB),
  * (DATA(76),FY), (DATA(258), $BT18)
  GO TO (10),I
10 IF (SE/ST.LE.201.0*SQRT(SKB/FY)) GO TO 100
  $BT18=0.0
  GO TO 102
100 $BT18=1.0
102 PRD(258)=.TRUE.
  RETURN
  END

```

FIG. 5.5 CONDITION AND ACTION SUBROUTINES FOR  
DECISION TABLE 13.5.A.1 (44)



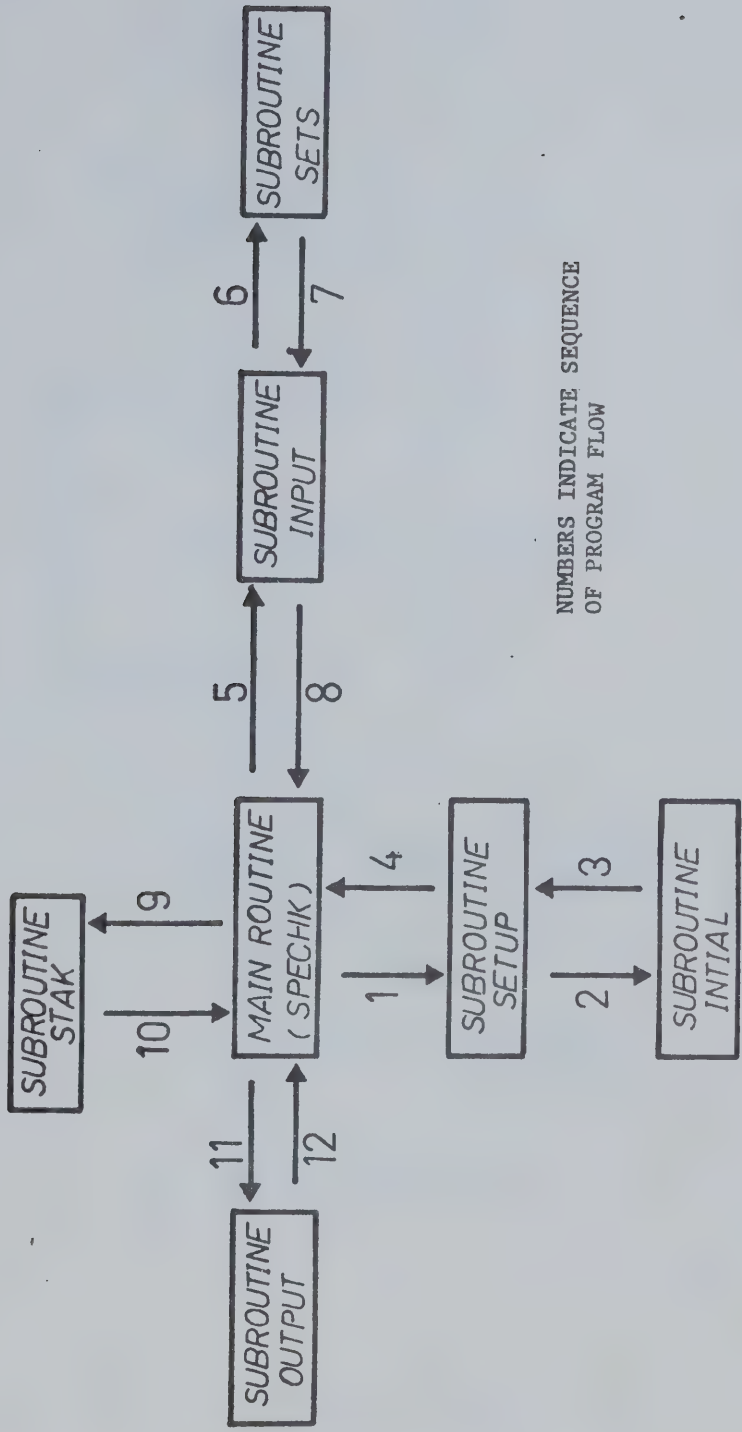


FIG. 7.1 SUBROUTINE STRUCTURE FOR BATCH MODE PROCESSING



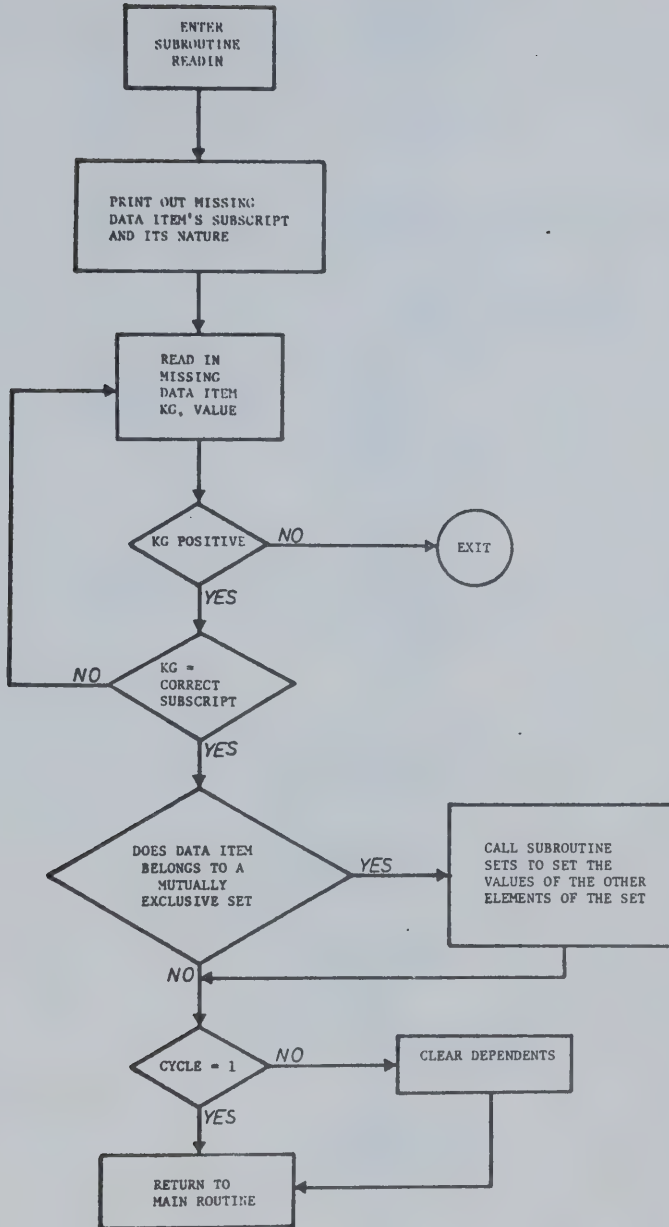


FIG. 7.2 MISSING DATA INPUT PROCEDURE IN INTERACTIVE MODE



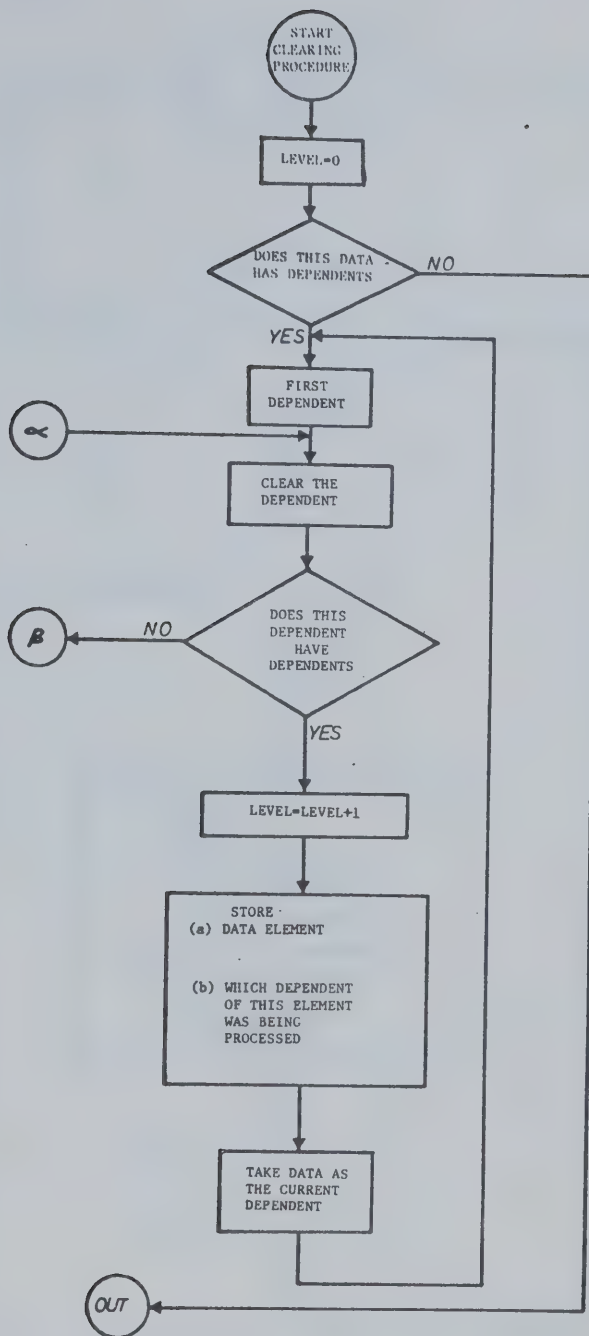


FIGURE 7.3 PROCEDURE FOR CLEARING DEPENDENT DATA ELEMENTS





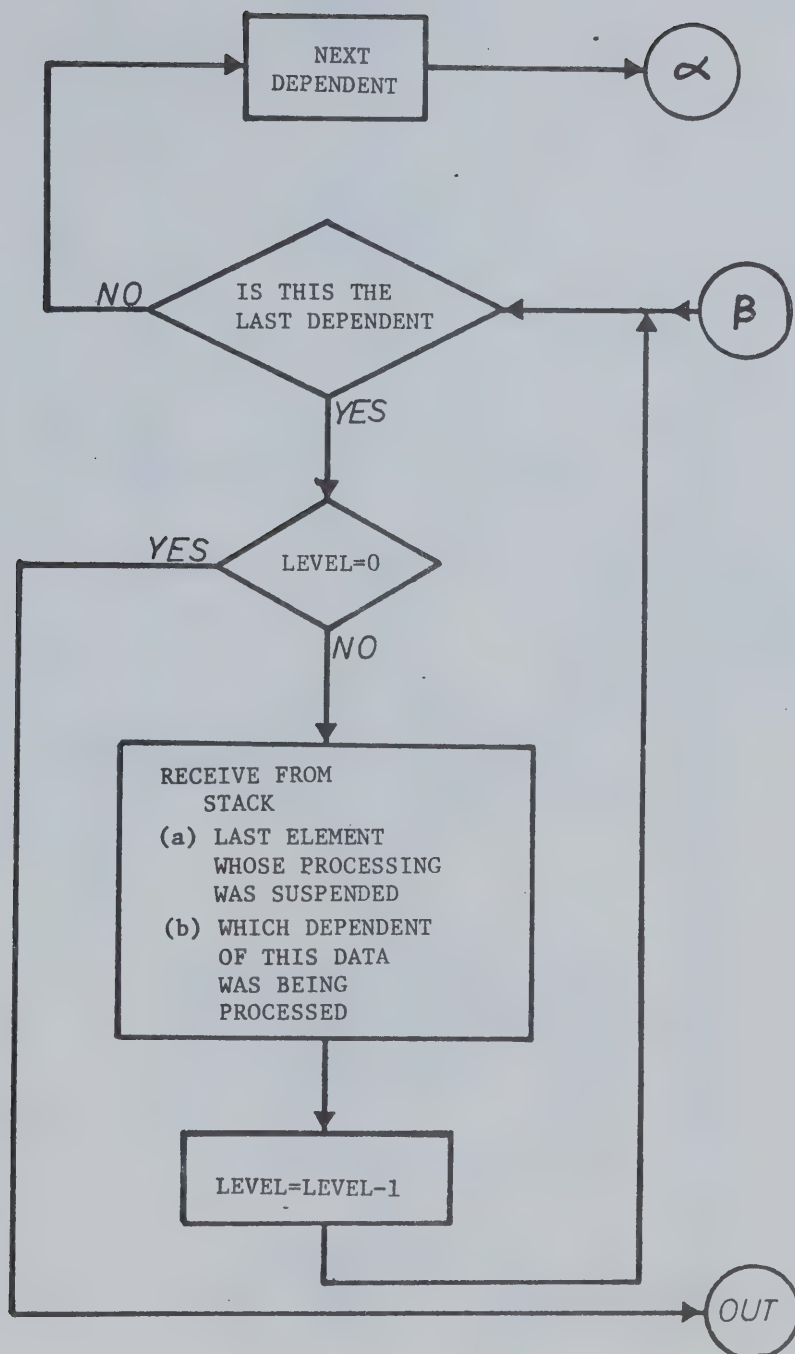


FIG. 7.4 PROCEDURE FOR CLEARING DEPENDENT DATA ELEMENTS



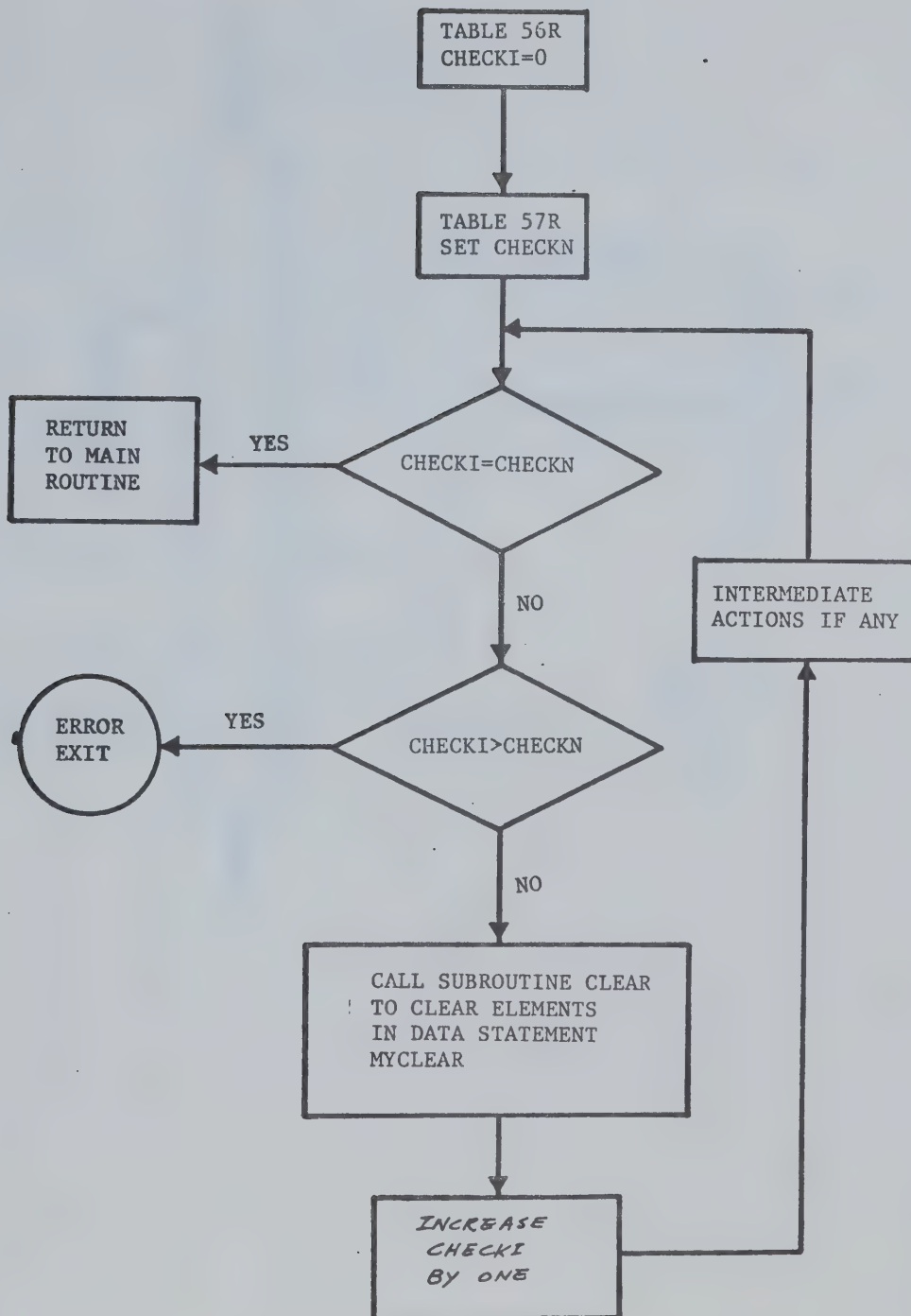


FIG. 7.5 RECURSIVE OPERATION OF A DECISION TABLE



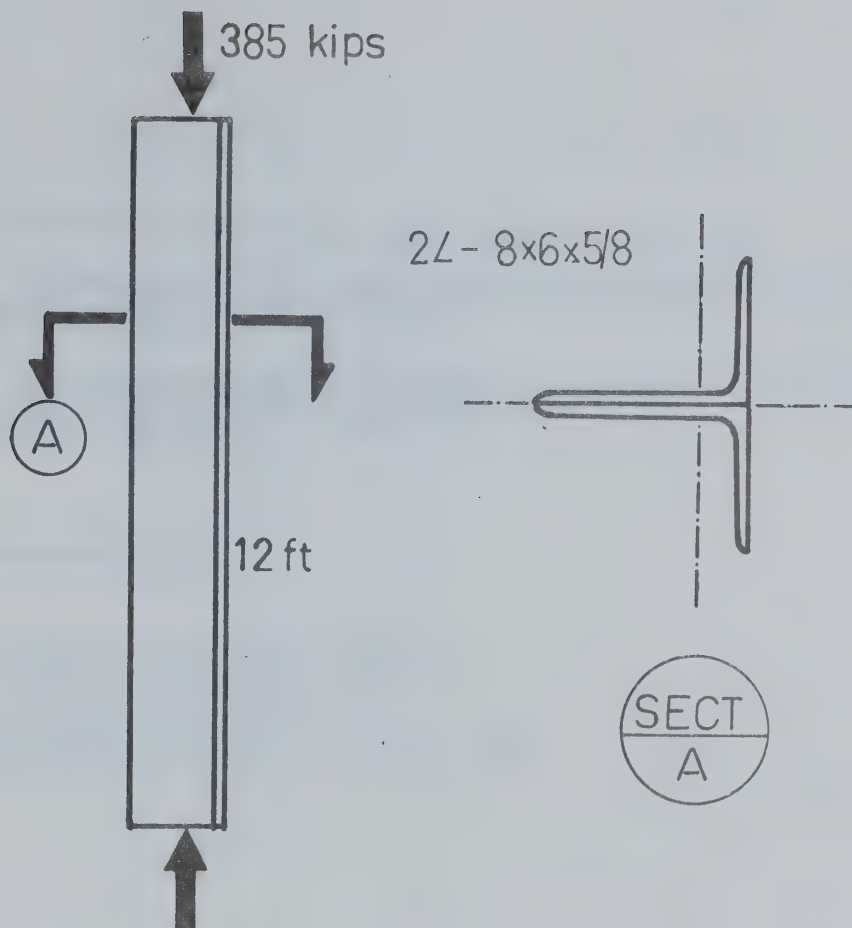


FIG. 8.1 LOADING CONDITION FOR EXAMPLE 1



```
*run obcombine 5=axcom 9=decldata1 8=csas16 2=mapnsave 6=savel 4=*source* 7=*sink*
#20:18.42
```

```
CONDITION NUMBER 50F TABLE 4IS NOT AVAILABLE. THIS CORRESPONDS TO DATA NUMBER 222
SUBROUTINE READIN IS CALLED TO INPUT THIS DATA ITEM
*****AWAITING INPUT FOR DATA ITEM WITH SUBSCRIPT= 222
THIS DATA ITEM IS A CONDITION.
222,0.0,
```

```
CONDITION NUMBER 60F TABLE 4IS NOT AVAILABLE. THIS CORRESPONDS TO DATA NUMBER 223
SUBROUTINE READIN IS CALLED TO INPUT THIS DATA ITEM
*****AWAITING INPUT FOR DATA ITEM WITH SUBSCRIPT= 223
THIS DATA ITEM IS A CONDITION.
223,0.0,
```

STRENGTH CRITERION SATISFIED

```
PLEASE INPUT A VALUE OF 1 OR 2 FOR INDIC
1 INDICATES THERE ARE FURTHER CYCLES
2 INDICATES NO FURTHER CYCLES
```

```
2,
EXECUTION OF PROGRAM IS COMPLETED.
COLLECT YOUR OUTPUT FROM THE COMPUTING CENTER.COME BACK SOON
#20:20.23 4.156 RC=0
#
```

FIG. 8.2 TERMINAL OUTPUT FOR EXAMPLE 1





THE FOLLOWING NUMERICAL DATA HAS BEEN SUPPLIED FOR CYCLE NUMBER 1

KGLOB	DATAK
1	1.0000
10	1.0000
23	1.0000
25	0.0
46	16.7000
58	0.6300
59	8.0000
76	44.0000
78	29000.0000
119	1.0000
121	0.0
122	1.0000
123	1.0000
124	1.0000
125	1.0000
166	1.0000
179	1.0000
186	144.0000
188	2.4200
197	0.9000
208	2.5400
216	144.0000
400	0.9000
401	1.2500
402	1.5000
403	1.5000
404	1.2500
405	1.0000
406	300.0000
407	85.0000
408	0.0
409	0.0
220	1.0000
221	1.0000

FIG. 8.3 COMPUTER OUTPUT FOR EXAMPLE 1



DATA PRINTED AGAIN FOR CHECKING. ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE

KGLOB	DATK	FRD			
1	1.0000	T			
2	0.0	T	401	1.2500	T
10	1.0000	T	402	1.5000	T
11	0.0	T	403	1.5000	T
12	0.0	T	404	1.2500	T
13	0.0	T	405	1.0000	T
14	0.0	T	406	300.0000	T
15	0.0	T	407	85.0000	T
16	0.0	T	408	0.0	T
17	0.0	T	409	0.0	T
23	1.0000	T			
24	0.0	T			
25	0.0	T			
46	16.7000	T			
58	0.6300	T			
59	8.0000	T			
76	44.0000	T			
78	29000.0000	T			
119	1.0000	T			
121	0.0	T			
122	1.0000	T			
123	1.0000	T			
124	1.0000	T			
125	1.0000	T			
126	0.0	T			
165	0.0	T			
166	1.0000	T			
167	0.0	T			
168	0.0	T			
169	0.0	T			
170	0.0	T			
171	0.0	T			
172	0.0	T			
173	0.0	T			
174	0.0	T			
175	0.0	T			
179	1.0000	T			
186	144.0000	T			
188	2.4200	T			
197	0.9000	T			
208	2.5400	T			
216	144.0000	T			
220	1.0000	T			
221	1.0000	T			
400	0.9000	T			

FIG. 8.4 COMPUTER OUTPUT FOR EXAMPLE 1



```

CYCLE NUMBER 1 *** START EXECUTION WITH TABLE 1 ***
  SCANNING OF TABLE 1 IS COMPLETE. RULE NO. 1 APPLIES
    SUSPENDED EXECUTION OF TABLE 1 AT ACTION 1 OF RULE 1
    STARTED EXECUTION OF TABLE 2 FOR DIRECT EXECUTION
      SCANNING OF TABLE 2 IS COMPLETE. RULE NO. 1 APPLIES
        SUSPENDED EXECUTION OF TABLE 2 AT ACTION 1 OF RULE 1
        STARTED EXECUTION OF TABLE 3 FOR DIRECT EXECUTION
          SCANNING OF TABLE 3 IS COMPLETE. RULE NO. 7 APPLIES
            SUSPENDED EXECUTION OF TABLE 3 AT ACTION 5 OF RULE 7
            STARTED EXECUTION OF TABLE 27 FOR DIRECT EXECUTION
              SUSPENDED EXECUTION OF TABLE 27 AT CONDITION 1 OF RULE 1
              REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 150
              STARTED EXECUTION OF TABLE 28
                SCANNING OF TABLE 28 IS COMPLETE. RULE NO. 1 APPLIES
                  SUSPENDED EXECUTION OF TABLE 28 AT ACTION 1 OF RULE 1
                  STARTED EXECUTION OF TABLE 30 FOR DIRECT EXECUTION
                    SUSPENDED EXECUTION OF TABLE 30 AT CONDITION 1 OF RULE 1
                    STARTED EXECUTION OF TABLE 5 TO OBTAIN VALUE OF DATA NUMBER 95
                      SCANNING OF TABLE 5 IS COMPLETE. RULE NO. 1 APPLIES
                        SUSPENDED EXECUTION OF TABLE 5 AT ACTION 1 OF RULE 1
                        STARTED EXECUTION OF TABLE 21 FOR DIRECT EXECUTION
                          SCANNING OF TABLE 21 IS COMPLETE. RULE NO. 11 APPLIES
                            RESTART EXECUTION OF TABLE 5 AT ACTION 1 OF RULE 1
                              RESTART EXECUTION OF TABLE 30 AT CONDITION 1 OF RULE 1
                                SCANNING OF TABLE 30 IS COMPLETE. RULE NO. 3 APPLIES
                                  SUSPENDED EXECUTION OF TABLE 30 AT ACTION 1 OF RULE 3
                                  REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 185
                                  STARTED EXECUTION OF TABLE 4
                                    SCANNING OF TABLE 4 IS COMPLETE. RULE NO. 4 APPLIES
                                      RESTART EXECUTION OF TABLE 30 AT ACTION 1 OF RULE 3
                                        SUSPENDED EXECUTION OF TABLE 30 AT ACTION 2 OF RULE 3
                                        STARTED EXECUTION OF TABLE 31 FOR DIRECT EXECUTION
                                          SCANNING OF TABLE 31 IS COMPLETE. RULE NO. 1 APPLIES
                                            SUSPENDED EXECUTION OF TABLE 31 AT ACTION 5 OF RULE 1
                                            REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 120
                                            STARTED EXECUTION OF TABLE 93
                                              SCANNING OF TABLE 93 IS COMPLETE. RULE NO. 7 APPLIES
                                                RESTART EXECUTION OF TABLE 31 AT ACTION 5 OF RULE 1
                                                  RESTART EXECUTION OF TABLE 30 AT ACTION 2 OF RULE 3
                                                    RESTART EXECUTION OF TABLE 28 AT ACTION 1 OF RULE 1
                                                      RESTART EXECUTION OF TABLE 27 AT CONDITION 1 OF RULE 1
                                                        SCANNING OF TABLE 27 IS COMPLETE. RULE NO. 1 APPLIES
STRENGTH CRITERION SATISFIED
  RESTART EXECUTION OF TABLE 3 AT ACTION 5 OF RULE 7
  RESTART EXECUTION OF TABLE 2 AT ACTION 1 OF RULE 1
  RESTART EXECUTION OF TABLE 1 AT ACTION 1 OF RULE 1

```

FIG. 8.5 COMPUTER OUTPUT FOR EXAMPLE 1



DATA VALUES AT THE END OF CYCLE NO. 1  
ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE

KGLOB	DATAK	PRD
1	1.0000	T
2	0.0	T
10	1.0000	T
11	0.0	T
12	0.0	T
13	0.0	T
14	0.0	T
15	0.0	T
16	0.0	T
17	0.0	T
23	1.0000	T
24	0.0	T
25	0.0	T
46	16.7000	T
58	0.6700	T
59	8.0000	T
76	44.0000	T
78	29000.0000	T
95	0.0	T
96	0.0	T
97	1.0000	T
98	0.0	T
119	1.0000	T
120	452.2498	T
121	0.0	T
122	1.0000	T
123	1.0000	T
124	1.0000	T
125	1.0000	T
126	0.0	T
145	505.6328	T
150	0.8944	T
151	0.0	T
152	0.0	T
153	0.0	T
154	1.0000	T
165	0.0	T
166	1.0000	T
167	0.0	T
168	0.0	T
169	0.0	T
170	0.0	T
171	0.0	T
172	0.0	T
173	0.0	T
174	0.0	T
175	0.0	T
179	1.0000	T
185	1.0000	T
186	144.0000	T
188	2.4200	T
197	0.9000	T
208	2.5400	T
214	1.0000	T
215	59.5041	T
216	144.0000	T
217	59.5041	T
219	56.6929	T
220	1.0000	T
221	1.0000	T
222	0.0	T
223	0.0	T
230	0.7382	T
231	1.0000	T
232	0.0	T
233	0.0	T
234	0.0	T
254	0.0	T
255	0.0	T
256	1.0000	T
257	0.0	T
400	0.9000	T
401	1.2500	T
402	1.5000	T
403	1.5000	T
404	1.2500	T
405	1.0000	T
406	300.0000	T
407	85.0000	T
408	0.0	T
409	0.0	T

EXECUTION OF PROGRAM IS COMPLETED

FIG. 8.6 COMPUTER OUTPUT FOR EXAMPLE 1





```
$run obcombine 5=lub 9=decidata1 8=csas16 2=mapnsave 6=save2 4==source* 7==sink*
#20:22.14
```

```
      ERROR MESSAGE; DATA NUMBER 63
IS NOT AVAILABLE.THIS IS AN INGREDIENT OF A CONDITION
*****AWAITING INPUT FOR DATA ITEM WITH SUBSCRIPT= 63
THIS DATA ITEM IS A MISSING INGREDIENT OF A CONDITION
63,10.91,
```

```
      STRENGTH CRITERION SATISFIED
*****SHEAR CRITERION SATISFIED*****
```

```
      PLEASE INPUT A VALUE OF 1 OR 2 FOR INDIC
1 INDICATES THERE ARE FURTHER CYCLES
2 INDICATES NO FURTHER CYCLES
```

```
2,
EXECUTION OF PROGRAM IS COMPLETED.
COLLECT YOUR OUTPUT FROM THE COMPUTING CENTER.COME BACK SOON
#20:23.35 5.243 RC=0
#
```

FIG. 8.7 TERMINAL OUTPUT FOR EXAMPLE 2



THE FOLLOWING NUMERICAL DATA HAS BEEN SUPPLIED FOR CYCLE NUMBER 1

KGLOB	DATAK
1	1.0000
10	1.0000
22	1.0000
26	1.0000
29	1.0000
40	1.0000
47	4.0500
48	5.1800
58	0.6410
59	4.0400
64	0.3710
65	12.1900
76	44.0000
78	29000.0000
79	72.5000
80	64.8000
119	1.0000
121	1.0000
122	0.0
123	1.0000
124	0.0
127	0.0
128	1.0000
129	0.0
130	1.0000
138	3190.0000
169	1.0000
176	0.0
177	1.0000
180	0.0
187	288.0000
189	2.2000
197	0.9000
225	1.0000
320	1.0000
342	1.0000
334	0.0
400	0.9000
401	1.2500
402	1.5000
403	1.5000
404	1.2500
405	1.0000
410	1036.8001
411	518.3999
412	0.0
413	0.0
418	14.4000
419	7.2000
420	0.0
421	0.0
125	0.0

FIG. 8.8 COMPUTER OUTPUT FOR EXAMPLE 2



DATA PRINTED AGAIN FOR CHECKING. ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE

KGLOB	DATAK	PED			
1	1.0000	T	169	1.0000	T
2	0.0	T	170	0.0	T
10	1.0000	T	171	0.0	T
11	0.0	T	172	0.0	T
12	0.0	T	173	0.0	T
13	0.0	T	174	0.0	T
14	0.0	T	175	0.0	T
15	0.0	T	176	0.0	T
16	0.0	T	177	1.0000	T
17	0.0	T	180	0.0	T
21	0.0	T	187	288.0000	T
22	1.0000	T	189	2.2000	T
26	1.0000	T	197	0.9000	T
27	0.0	T	224	0.0	T
28	0.0	T	225	1.0000	T
29	1.0000	T	226	0.0	T
30	0.0	T	320	1.0000	T
40	1.0000	T	321	0.0	T
41	0.0	T	322	0.0	T
47	4.0500	T	334	0.0	T
48	5.1800	T	342	1.0000	T
58	0.6410	T	343	0.0	T
59	4.0400	T	344	0.0	T
64	0.3710	T	400	0.9000	T
65	12.1900	T	401	1.2500	T
76	44.0000	T	402	1.5000	T
78	29000.0000	T	403	1.5000	T
79	72.5000	T	404	1.2500	T
80	64.8000	T	405	1.0000	T
119	1.0000	T	410	1036.8001	T
121	1.0000	T	411	518.3999	T
122	0.0	T	412	0.0	T
123	1.0000	T	413	0.0	T
124	0.0	T	418	14.4000	T
125	0.0	T	419	7.2000	T
126	0.0	T	420	0.0	T
127	0.0	T	421	0.0	T
128	1.0000	T			
129	0.0	T			
130	1.0000	T			
138	3190.0000	T			
165	0.0	T			
166	0.0	T			
167	0.0	T			
168	0.0	T			

FIG. 8.9 COMPUTER OUTPUT FOR EXAMPLE 2



```

CYCLE NUMBER 1    ***  START EXECUTION WITH TABLE  1  ***

  SCANNING OF TABLE  1 IS COMPLETE. RULE NO.  1 APPLIES
SUSPENDED EXECUTION OF TABLE  1 AT ACTION  1 OF RULE  1
STARTED EXECUTION OF TABLE  2 FOR DIRECT EXECUTION

  SCANNING OF TABLE  2 IS COMPLETE. RULE NO.  1 APPLIES
SUSPENDED EXECUTION OF TABLE  2 AT ACTION  1 OF RULE  1
STARTED EXECUTION OF TABLE  3 FOR DIRECT EXECUTION

  SCANNING OF TABLE  3 IS COMPLETE. RULE NO.  5 APPLIES
SUSPENDED EXECUTION OF TABLE  3 AT ACTION  5 OF RULE  5
STARTED EXECUTION OF TABLE  27 FOR DIRECT EXECUTION

SUSPENDED EXECUTION OF TABLE  27 AT CONDITION  1 OF RULE  1
REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 151
STARTED EXECUTION OF TABLE  42

  SCANNING OF TABLE  42 IS COMPLETE. RULE NO.  2 APPLIES
SUSPENDED EXECUTION OF TABLE  42 AT ACTION  3 OF RULE  2
STARTED EXECUTION OF TABLE  46 FOR DIRECT EXECUTION

  SCANNING OF TABLE  46 IS COMPLETE. RULE NO.  1 APPLIES
SUSPENDED EXECUTION OF TABLE  46 AT ACTION  2 OF RULE  1
STARTED EXECUTION OF TABLE  47 FOR DIRECT EXECUTION

SUSPENDED EXECUTION OF TABLE  47 AT CONDITION  3 OF RULE  1
STARTED EXECUTION OF TABLE  5 TO OBTAIN VALUE OF DATA NUMBER  95

SUSPENDED EXECUTION OF TABLE  5 AT CONDITION  7 OF RULE  7
STARTED EXECUTION OF TABLE  6 TO OBTAIN VALUE OF DATA NUMBER  99

  SCANNING OF TABLE  6 IS COMPLETE. RULE NO.  2 APPLIES
SUSPENDED EXECUTION OF TABLE  6 AT ACTION  1 OF RULE  2
STARTED EXECUTION OF TABLE  14 FOR DIRECT EXECUTION

  SCANNING OF TABLE  14 IS COMPLETE. RULE NO.  2 APPLIES
SUSPENDED EXECUTION OF TABLE  14 AT ACTION  2 OF RULE  2
STARTED EXECUTION OF TABLE  16 FOR DIRECT EXECUTION

  SCANNING OF TABLE  16 IS COMPLETE. RULE NO.  1 APPLIES
RESTART EXECUTION OF TABLE  14  AT ACTION  2  OF RULE  2

RESTART EXECUTION OF TABLE  6  AT ACTION  1  OF RULE  2
RESTART EXECUTION OF TABLE  5  AT CONDITION  7  OF RULE  7
SUSPENDED EXECUTION OF TABLE  5 AT CONDITION 11 OF RULE  7
STARTED EXECUTION OF TABLE  20 TO OBTAIN VALUE OF DATA NUMBER 110

  SCANNING OF TABLE  20 IS COMPLETE. RULE NO.  2 APPLIES
SUSPENDED EXECUTION OF TABLE  20 AT ACTION  1 OF RULE  2
STARTED EXECUTION OF TABLE  21 FOR DIRECT EXECUTION

  SCANNING OF TABLE  21 IS COMPLETE. RULE NO.  5 APPLIES

```

FIG. 8.10 COMPUTER OUTPUT FOR EXAMPLE 2





RESTART EXECUTION OF TABLE 20 AT ACTION 1 OF RULE 2  
 RESTART EXECUTION OF TABLE 5 AT CONDITION 11 OF RULE 7  
 SCANNING OF TABLE 5 IS COMPLETE. RULE NO. 7 APPLIES  
 SUSPENDED EXECUTION OF TABLE 5 AT ACTION 7 OF RULE 7  
 STARTED EXECUTION OF TABLE 7 FOR DIRECT EXECUTION  
 SCANNING OF TABLE 7 IS COMPLETE. RULE NO. 4 APPLIES  
 RESTART EXECUTION OF TABLE 5 AT ACTION 7 OF RULE 7  
 RESTART EXECUTION OF TABLE 47 AT CONDITION 3 OF RULE 1  
 SCANNING OF TABLE 47 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 47 AT ACTION 1 OF RULE 1  
 STARTED EXECUTION OF TABLE 48 FOR DIRECT EXECUTION  
 SUSPENDED EXECUTION OF TABLE 48 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 137  
 STARTED EXECUTION OF TABLE 51  
 SCANNING OF TABLE 51 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 51 AT ACTION 2 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 194  
 STARTED EXECUTION OF TABLE 52  
 SCANNING OF TABLE 52 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 52 AT ACTION 1 OF RULE 1  
 STARTED EXECUTION OF TABLE 53 FOR DIRECT EXECUTION  
 SCANNING OF TABLE 53 IS COMPLETE. RULE NO. 2 APPLIES  
  
 RESTART EXECUTION OF TABLE 52 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 51 AT ACTION 2 OF RULE 1  
 SUSPENDED EXECUTION OF TABLE 51 AT ACTION 2 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 195  
 STARTED EXECUTION OF TABLE 55  
 SCANNING OF TABLE 55 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 51 AT ACTION 2 OF RULE 1  
 RESTART EXECUTION OF TABLE 48 AT CONDITION 1 OF RULE 1  
 SCANNING OF TABLE 48 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 48 AT ACTION 3 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 132  
 STARTED EXECUTION OF TABLE 93  
 SCANNING OF TABLE 93 IS COMPLETE. RULE NO. 5 APPLIES  
 RESTART EXECUTION OF TABLE 48 AT ACTION 3 OF RULE 1  
 RESTART EXECUTION OF TABLE 47 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 46 AT ACTION 2 OF RULE 1  
 RESTART EXECUTION OF TABLE 42 AT ACTION 3 OF RULE 2  
 RESTART EXECUTION OF TABLE 27 AT CONDITION 1 OF RULE 1  
 SCANNING OF TABLE 27 IS COMPLETE. RULE NO. 1 APPLIES

FIG. 8.11 COMPUTER OUTPUT FOR EXAMPLE 2



STRENGTH CRITERION SATISFIED

RESTART EXECUTION OF TABLE 3 AT ACTION 5 OF RULE 5

SUSPENDED EXECUTION OF TABLE 3 AT ACTION 7 OF RULE 5  
 STARTED EXECUTION OF TABLE 74 FOR DIRECT EXECUTION

SUSPENDED EXECUTION OF TABLE 74 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 326  
 STARTED EXECUTION OF TABLE 75

SCANNING OF TABLE 75 IS COMPLETE. RULE NO. 1 APPLIES

SUSPENDED EXECUTION OF TABLE 75 AT ACTION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 327  
 STARTED EXECUTION OF TABLE 76

SUSPENDED EXECUTION OF TABLE 76 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 328  
 STARTED EXECUTION OF TABLE 77

SCANNING OF TABLE 77 IS COMPLETE. RULE NO. 1 APPLIES

RESTART EXECUTION OF TABLE 76 AT CONDITION 1 OF RULE 1

SCANNING OF TABLE 76 IS COMPLETE. RULE NO. 1 APPLIES

RESTART EXECUTION OF TABLE 75 AT ACTION 1 OF RULE 1

RESTART EXECUTION OF TABLE 74 AT CONDITION 1 OF RULE 1

SCANNING OF TABLE 74 IS COMPLETE. RULE NO. 1 APPLIES

\*\*\*\*\*SHEAR CRITERION SATISFIED\*\*\*\*\*

RESTART EXECUTION OF TABLE 3 AT ACTION 7 OF RULE 5

RESTART EXECUTION OF TABLE 2 AT ACTION 1 OF RULE 1

RESTART EXECUTION OF TABLE 1 AT ACTION 1 OF RULE 1

FIG. 8.12 COMPUTER OUTPUT FOR EXAMPLE 2



DATA VALUES AT THE END OF CYCLE NO. 1  
 ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE

KGLOB	DATAK	PRD			
1	1.0000	T	165	0.0	T
2	0.0	T	166	0.0	T
10	1.0000	T	167	0.0	T
11	0.0	T	168	0.0	T
12	0.0	T	169	1.0000	T
13	0.0	T	170	0.0	T
14	0.0	T	171	0.0	T
15	0.0	T	172	0.0	T
16	0.0	T	173	0.0	T
17	0.0	T	174	0.0	T
21	0.0	T	175	0.0	T
22	1.0000	T	176	0.0	T
26	1.0000	T	177	1.0000	T
27	0.0	T	180	0.0	T
28	0.0	T	187	288.0000	T
29	1.0000	T	189	2.2000	T
30	0.0	T	194	1.0000	T
40	1.0000	T	195	29.5096	T
41	0.0	T	196	14.5881	T
47	4.0500	T	197	0.9000	T
48	5.1800	T	224	0.0	T
58	0.6410	T	225	1.0000	T
59	4.0400	T	226	0.0	T
63	10.9100	T	254	1.0000	T
64	0.3710	T	255	0.0	T
65	12.1900	T	256	0.0	T
76	44.0000	T	257	0.0	T
78	29000.0000	T	262	1.0000	T
79	72.5000	T	263	0.0	T
80	64.8000	T	264	0.0	T
95	1.0000	T	265	0.0	T
96	0.0	T	275	0.0	T
97	0.0	T	284	1.0000	T
98	0.0	T	285	0.0	T
99	1.0000	T	286	0.0	T
100	0.0	T	287	0.0	T
101	0.0	T	320	1.0000	T
102	0.0	T	321	0.0	T
110	1.0000	T	322	0.0	T
111	0.0	T	325	25.9200	T
112	0.0	T	326	105.8508	T
113	0.0	T	327	29.0400	T
119	1.0000	T	328	5.3400	T
121	1.0000	T	329	1.0000	T
122	0.0	T	330	0.0	T
123	1.0000	T	334	0.0	T
124	0.0	T	342	1.0000	T
125	0.0	T	343	0.0	T
126	0.0	T	344	0.0	T
127	0.0	T	400	0.9000	T
128	1.0000	T	401	1.2500	T
129	0.0	T	402	1.5000	T
130	1.0000	T	403	1.5000	T
132	1866.2398	T	404	1.2500	T
134	1919.1516	T	405	1.0000	T
137	2133.1218	T	410	1036.8001	T
138	3190.0000	T	411	518.3999	T
139	1.0000	T	412	0.0	T
150	0.0	T	413	0.0	T
151	0.9724	T	418	14.4000	T
152	0.0	T	419	7.2000	T
154	1.0000	T	420	0.0	T
155	1.0000	T	421	0.0	T

EXECUTION OF PROGRAM IS COMPLETED

FIG. 8.13 COMPUTER OUTPUT FOR EXAMPLE 2



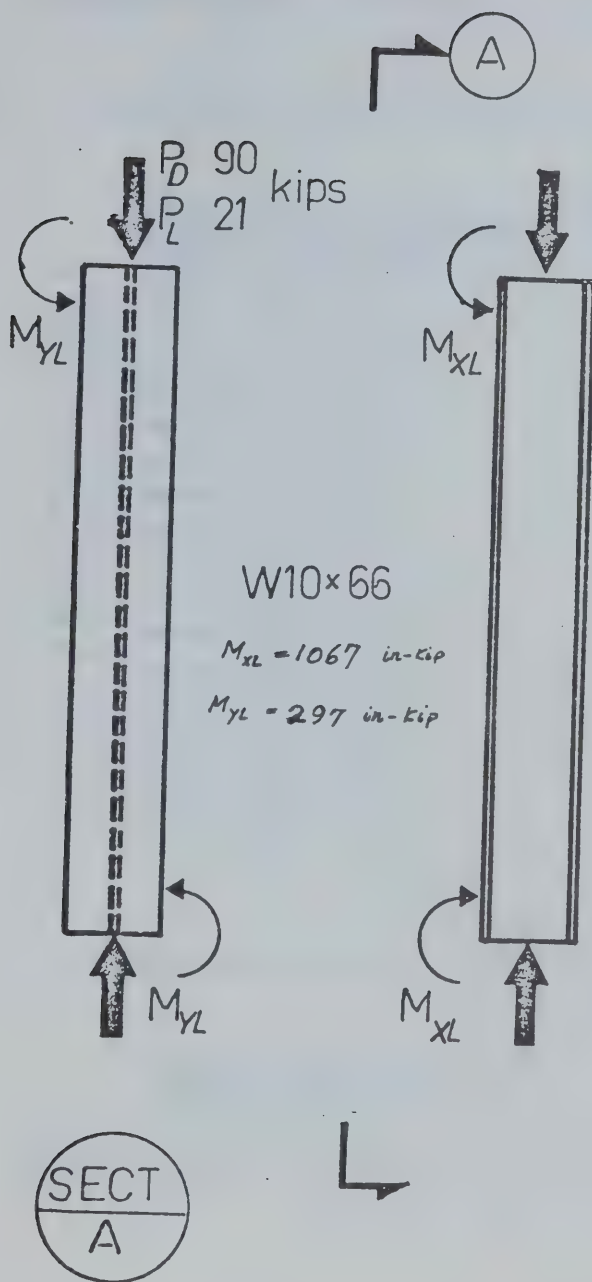


FIG. 8.14 LOADING CONDITION FOR EXAMPLE 3





```
$run obcombine 5=comben 9=decidatal 8=csas16 2=mapsave 6=save 4=source 7=sink
#21:49:35
```

\*\*\*\*\* BOTH STRENGTH AND STABILITY CRITERIA NOT SATISFIED \*\*\*\*\*

```
PLEASE INPUT A VALUE OF 1 OR 2 FOR INDIC
1 INDICATES THERE ARE FURTHER CYCLES
2 INDICATES NO FURTHER CYCLES

1,
65,10,38, AWAITING NEXT DATA ITEM
59,5.06, AWAITING NEXT DATA ITEM
58,0.748, AWAITING NEXT DATA ITEM
64,0.457, AWAITING NEXT DATA ITEM
63,8.88, AWAITING NEXT DATA ITEM
79,82.8, AWAITING NEXT DATA ITEM
80,73.6, AWAITING NEXT DATA ITEM
46,19.4, AWAITING NEXT DATA ITEM
48,7.57, AWAITING NEXT DATA ITEM
188,2.58, AWAITING NEXT DATA ITEM
208,4.44, AWAITING NEXT DATA ITEM
189,2.8, AWAITING NEXT DATA ITEM
138,3646.27, AWAITING NEXT DATA ITEM
190,853.5, AWAITING NEXT DATA ITEM
440,25.5, AWAITING NEXT DATA ITEM
441,38.8, AWAITING NEXT DATA ITEM
0,
```

\*\*\*\*\* STRENGTH AND STABILITY CRITERIA SATISFIED.\*\*\*\*\*

```
PLEASE INPUT A VALUE OF 1 OR 2 FOR INDIC
1 INDICATES THERE ARE FURTHER CYCLES
2 INDICATES NO FURTHER CYCLES

2,
EXECUTION OF PROGRAM IS COMPLETED.
COLLECT YOUR OUTPUT FROM THE COMPUTING CENTER.COME BACK SOON
#21:54:04 1.048 RC=0
#
```

FIG. 8.15 TERMINAL OUTPUT FOR EXAMPLE 3



THE FOLLOWING NUMERICAL DATA HAS BEEN SUPPLIED FOR CYCLE NUMBER 1

KGLOB	DATAK		
1	1.0000	404	1.2500
10	1.0000	405	1.0000
23	1.0000	406	90.0000
25	0.0	407	21.0000
26	1.0000	408	0.0
28	1.0000	409	0.0
40	1.0000	410	0.0
46	14.4000	411	1067.0000
48	5.5800	412	0.0
58	0.5580	413	0.0
59	5.0000	414	0.0
63	8.8800	415	297.0000
64	0.3400	416	0.0
76	44.0000	417	0.0
78	2900.0000	186	114.0000
79	60.3000	216	114.0000
80	54.6000	180	0.0
119	1.0000	130	1.0000
121	0.0	178	1.0000
122	0.0	222	0.0
123	0.0	223	0.0
124	1.0000	176	0.0
125	1.0000	65	10.0000
127	1.0000	323	0.0
128	1.0000	324	1.0000
129	1.0000	440	18.6000
138	2490.3899	441	28.2000
169	1.0000		
177	1.0000		
187	114.0000		
188	2.5400		
189	2.7700		
190	633.6001		
197	0.9000		
208	4.3500		
220	1.0000		
221	1.0000		
226	1.0000		
290	1.0000		
320	1.0000		
342	1.0000		
400	0.9000		
401	1.2500		
402	1.5000		
403	1.5000		

FIG. 8.16 COMPUTER OUTPUT FOR EXAMPLE 3



DATA PRINTED AGAIN FOR CHECKING. ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE

KGLOB	DATAK	PRD			
1	1.0000	T	197	0.9000	T
2	0.0	T	208	4.3500	T
10	1.0000	T	216	114.0000	T
11	0.0	T	220	1.0000	T
12	0.0	T	221	1.0000	T
13	0.0	T	222	0.0	T
14	0.0	T	223	0.0	T
15	0.0	T	224	0.0	T
16	0.0	T	225	0.0	T
17	0.0	T	226	1.0000	T
23	1.0000	T	290	1.0000	T
24	0.0	T	291	0.0	T
25	0.0	T	320	1.0000	T
26	1.0000	T	321	0.0	T
27	0.0	T	322	0.0	T
28	1.0000	T	323	0.0	T
29	0.0	T	324	1.0000	T
30	0.0	T	342	1.0000	T
40	1.0000	T	343	0.0	T
41	0.0	T	344	0.0	T
46	14.4000	T	400	0.9000	T
48	5.5800	T	401	1.2500	T
58	0.5580	T	402	1.5000	T
59	5.0000	T	403	1.5000	T
63	8.8800	T	404	1.2500	T
64	0.3400	T	405	1.0000	T
65	10.0000	T	406	90.0000	T
76	44.0000	T	407	21.0000	T
78	29000.0000	T	408	0.0	T
79	60.3000	T	409	0.0	T
80	54.6000	T	410	0.0	T
119	1.0000	T	411	1067.0000	T
121	0.0	T	412	0.0	T
122	0.0	T	413	0.0	T
123	0.0	T	414	0.0	T
124	1.0000	T	415	297.0000	T
125	1.0000	T	416	0.0	T
126	0.0	T	417	0.0	T
127	1.0000	T	440	18.6000	T
128	1.0000	T	441	28.2000	T
129	1.0000	T			
130	1.0000	T			
138	2490.3899	T			
165	0.0	T			
166	0.0	T			
167	0.0	T			
168	0.0	T			
169	1.0000	T			
170	0.0	T			
171	0.0	T			
172	0.0	T			
173	0.0	T			
174	0.0	T			
175	0.0	T			
176	0.0	T			
177	1.0000	T			
178	1.0000	T			
180	0.0	T			
186	114.0000	T			
187	114.0000	T			
188	2.5400	T			
189	2.7700	T			
190	633.6001	T			

FIG. 8.17 COMPUTER OUTPUT FOR EXAMPLE 3



```

CYCLE NUMBER 1      ***  START EXECUTION WITH TABLE 1  ***

  SCANNING OF TABLE 1 IS COMPLETE. RULE NO. 1 APPLIES
SUSPENDED EXECUTION OF TABLE 1 AT ACTION 1 OF RULE 1
  STARTED EXECUTION OF TABLE 2 FOR DIRECT EXECUTION

  SCANNING OF TABLE 2 IS COMPLETE. RULE NO. 1 APPLIES
SUSPENDED EXECUTION OF TABLE 2 AT ACTION 1 OF RULE 1
  STARTED EXECUTION OF TABLE 3 FOR DIRECT EXECUTION

  SCANNING OF TABLE 3 IS COMPLETE. RULE NO. 13 APPLIES
SUSPENDED EXECUTION OF TABLE 3 AT ACTION 6 OF RULE 13
  STARTED EXECUTION OF TABLE 56 FOR DIRECT EXECUTION

  SCANNING OF TABLE 56 IS COMPLETE. RULE NO. 1 APPLIES
SUSPENDED EXECUTION OF TABLE 56 AT ACTION 1 OF RULE 1
  STARTED EXECUTION OF TABLE 57 FOR DIRECT EXECUTION

SUSPENDED EXECUTION OF TABLE 57 AT CONDITION 3 OF RULE 1
  STARTED EXECUTION OF TABLE 5 TO OBTAIN VALUE OF DATA NUMBER 95

SUSPENDED EXECUTION OF TABLE 5 AT CONDITION 7 OF RULE 7
  STARTED EXECUTION OF TABLE 6 TO OBTAIN VALUE OF DATA NUMBER 99

  SCANNING OF TABLE 6 IS COMPLETE. RULE NO. 2 APPLIES
SUSPENDED EXECUTION OF TABLE 6 AT ACTION 1 OF RULE 2
  STARTED EXECUTION OF TABLE 14 FOR DIRECT EXECUTION

  SCANNING OF TABLE 14 IS COMPLETE. RULE NO. 3 APPLIES
SUSPENDED EXECUTION OF TABLE 14 AT ACTION 3 OF RULE 3
  STARTED EXECUTION OF TABLE 17 FOR DIRECT EXECUTION

SUSPENDED EXECUTION OF TABLE 17 AT CONDITION 1 OF RULE 1
REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 120
  STARTED EXECUTION OF TABLE 93

  SCANNING OF TABLE 93 IS COMPLETE. RULE NO. 13 APPLIES
  RESTART EXECUTION OF TABLE 17 AT CONDITION 1 OF RULE 1

  SCANNING OF TABLE 17 IS COMPLETE. RULE NO. 1 APPLIES
SUSPENDED EXECUTION OF TABLE 17 AT ACTION 1 OF RULE 1
  STARTED EXECUTION OF TABLE 18 FOR DIRECT EXECUTION

  SCANNING OF TABLE 18 IS COMPLETE. RULE NO. 1 APPLIES
  RESTART EXECUTION OF TABLE 17 AT ACTION 1 OF RULE 1
  RESTART EXECUTION OF TABLE 14 AT ACTION 3 OF RULE 3
  RESTART EXECUTION OF TABLE 6 AT ACTION 1 OF RULE 2
  RESTART EXECUTION OF TABLE 5 AT CONDITION 7 OF RULE 7

SUSPENDED EXECUTION OF TABLE 5 AT CONDITION 11 OF RULE 7
  STARTED EXECUTION OF TABLE 20 TO OBTAIN VALUE OF DATA NUMBER 110

  SCANNING OF TABLE 20 IS COMPLETE. RULE NO. 2 APPLIES
SUSPENDED EXECUTION OF TABLE 20 AT ACTION 1 OF RULE 2
  STARTED EXECUTION OF TABLE 21 FOR DIRECT EXECUTION

  SCANNING OF TABLE 21 IS COMPLETE. RULE NO. 6 APPLIES
  RESTART EXECUTION OF TABLE 20 AT ACTION 1 OF RULE 2
  RESTART EXECUTION OF TABLE 5 AT CONDITION 11 OF RULE 7

  SCANNING OF TABLE 5 IS COMPLETE. RULE NO. 8 APPLIES

```

FIG. 8.18 COMPUTER OUTPUT FOR EXAMPLE 3





SUSPENDED EXECUTION OF TABLE 5 AT ACTION 8 OF RULE 8  
 STARTED EXECUTION OF TABLE 8 FOR DIRECT EXECUTION  
 SCANNING OF TABLE 8 IS COMPLETE. RULE NO. 5 APPLIES  
 RESTART EXECUTION OF TABLE 5 AT ACTION 8 OF RULE 8  
 RESTART EXECUTION OF TABLE 57 AT CONDITION 3 OF RULE 1  
 SCANNING OF TABLE 57 IS COMPLETE. RULE NO. 2 APPLIES  
 SUSPENDED EXECUTION OF TABLE 57 AT ACTION 1 OF RULE 2  
 STARTED EXECUTION OF TABLE 58 FOR DIRECT EXECUTION  
 SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 295  
 STARTED EXECUTION OF TABLE 60  
 SCANNING OF TABLE 60 IS COMPLETE. RULE NO. 2 APPLIES  
 RESTART EXECUTION OF TABLE 58 AT CONDITION 1 OF RULE 1  
 SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 2 OF RULE 4  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 298  
 STARTED EXECUTION OF TABLE 67  
 SCANNING OF TABLE 67 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 58 AT CONDITION 2 OF RULE 4  
 SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 297  
 STARTED EXECUTION OF TABLE 65  
 SUSPENDED EXECUTION OF TABLE 65 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 313  
 STARTED EXECUTION OF TABLE 66  
 SCANNING OF TABLE 66 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 66 AT ACTION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 215  
 STARTED EXECUTION OF TABLE 4  
 SCANNING OF TABLE 4 IS COMPLETE. RULE NO. 4 APPLIES  
 RESTART EXECUTION OF TABLE 66 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 65 AT CONDITION 1 OF RULE 1  
 SCANNING OF TABLE 65 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 194  
 STARTED EXECUTION OF TABLE 52  
 SCANNING OF TABLE 52 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 52 AT ACTION 1 OF RULE 1  
 STARTED EXECUTION OF TABLE 53 FOR DIRECT EXECUTION  
 SCANNING OF TABLE 53 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 53 AT ACTION 1 OF RULE 1  
 STARTED EXECUTION OF TABLE 54 FOR DIRECT EXECUTION  
 SCANNING OF TABLE 54 IS COMPLETE. RULE NO. 3 APPLIES  
 RESTART EXECUTION OF TABLE 53 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 52 AT ACTION 1 OF RULE 1

FIG. 8.19 COMPUTER OUTPUT FOR EXAMPLE 3



RESTART EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 299  
 STARTED EXECUTION OF TABLE 62  
 SCANNING OF TABLE 62 IS COMPLETE. RULE NO. 2 APPLIES  
 SUSPENDED EXECUTION OF TABLE 62 AT ACTION 1 OF RULE 2  
 STARTED EXECUTION OF TABLE 63 FOR DIRECT EXECUTION  
 SUSPENDED EXECUTION OF TABLE 63 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 137  
 STARTED EXECUTION OF TABLE 51  
 SCANNING OF TABLE 51 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 51 AT ACTION 2 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 195  
 STARTED EXECUTION OF TABLE 55  
 SCANNING OF TABLE 55 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 51 AT ACTION 2 OF RULE 1  
 RESTART EXECUTION OF TABLE 63 AT CONDITION 1 OF RULE 1  
 SCANNING OF TABLE 63 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 62 AT ACTION 1 OF RULE 2  
 RESTART EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 311  
 STARTED EXECUTION OF TABLE 68  
 SCANNING OF TABLE 68 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 310  
 STARTED EXECUTION OF TABLE 69  
 SCANNING OF TABLE 69 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 69 AT ACTION 1 OF RULE 1  
 STARTED EXECUTION OF TABLE 70 FOR DIRECT EXECUTION  
 SCANNING OF TABLE 70 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 70 AT ACTION 1 OF RULE 1  
 STARTED EXECUTION OF TABLE 71 FOR DIRECT EXECUTION  
 SCANNING OF TABLE 71 IS COMPLETE. RULE NO. 4 APPLIES  
 RESTART EXECUTION OF TABLE 70 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 69 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 5  
 SCANNING OF TABLE 58 IS COMPLETE. RULE NO. 6 APPLIES  
 \*\*\*\*\* BOTH STRENGTH AND STABILITY CRITERIA NOT SATISFIED \*\*\*\*\*  
 RESTART EXECUTION OF TABLE 57 AT ACTION 1 OF RULE 2  
 RESTART EXECUTION OF TABLE 56 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 3 AT ACTION 6 OF RULE 13  
 RESTART EXECUTION OF TABLE 2 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 1 AT ACTION 1 OF RULE 1

FIG. 8.20 COMPUTER OUTPUT FOR EXAMPLE 3



DATA VALUES AT THE END OF CYCLE NO. 1  
ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE

KGLOB	DATAK	PRD			
			178	1.0000	T
			180	0.0	T
			185	1.0000	T
			186	114.0000	T
			187	114.0000	T
			188	2.5400	T
			189	2.7700	T
			190	633.6001	T
			194	1.0000	T
			195	97.8947	T
			196	147.6012	T
			197	0.9000	T
			208	4.3500	T
			214	1.0000	T
			215	44.8819	T
			216	114.0000	T
			217	44.8819	T
			219	26.2069	T
			220	1.0000	T
			221	1.0000	T
			222	0.0	T
			223	0.0	T
			224	0.0	T
			225	0.0	T
			226	1.0000	T
			254	0.0	T
			255	1.0000	T
			256	0.0	T
			257	0.0	T
			266	1.0000	T
			267	0.0	T
			268	0.0	T
			275	0.0	T
			290	1.0000	T
			291	0.0	T
			295	2241.3508	T
			296	1116.7195	T
			297	486.7878	T
			298	570.2397	T
			299	2241.3508	T
			310	0.8500	T
			311	5996.4922	T
			312	2044.4961	T
			313	0.5568	T
			314	1.0000	T
			315	0.0	T
			316	0.0	T
			317	0.0	T
			320	1.0000	T
			321	0.0	T
			322	0.0	T
			323	0.0	T
			324	1.0000	T
			335	0.0	T
			336	1.0000	T
			337	0.0	T
			342	1.0000	T
			343	0.0	T
			344	0.0	T
			400	0.9000	T
			401	1.2500	T
			402	1.5000	T
			403	1.5000	T
			404	1.2500	T
			405	1.0000	T
			406	90.0000	T
			407	21.0000	T
			408	0.0	T
			409	0.0	T
			410	0.0	T
			411	1067.0000	T
			412	0.0	T
			413	0.0	T
			414	0.0	T
			415	297.0000	T
			416	0.0	T
			417	0.0	T
			440	18.6000	T
			441	28.2000	T
1	1.0000	T			
2	0.0	T			
10	1.0000	T			
11	0.0	T			
12	0.0	T			
13	0.0	T			
14	0.0	T			
15	0.0	T			
16	0.0	T			
17	0.0	T			
23	1.0000	T			
24	0.0	T			
25	0.0	T			
26	1.0000	T			
27	0.0	T			
28	1.0000	T			
29	0.0	T			
30	0.0	T			
40	1.0000	T			
41	0.0	T			
46	14.4000	T			
48	5.5800	T			
58	0.5580	T			
59	5.0000	T			
63	8.8800	T			
64	0.3400	T			
65	10.0000	T			
76	44.0000	T			
78	29000.0000	T			
79	60.3000	T			
80	54.6000	T			
95	0.0	T			
96	1.0000	T			
97	0.0	T			
98	0.0	T			
99	1.0000	T			
100	0.0	T			
101	0.0	T			
102	0.0	T			
110	0.0	T			
111	1.0000	T			
112	0.0	T			
113	0.0	T			
119	1.0000	T			
120	129.6000	T			
121	0.0	T			
122	0.0	T			
123	0.0	T			
124	1.0000	T			
125	1.0000	T			
126	0.0	T			
127	1.0000	T			
128	1.0000	T			
129	1.0000	T			
130	1.0000	T			
132	1440.4500	T			
133	400.9500	T			
136	1.0000	T			
137	9670.4375	T			
138	2490.3899	T			
139	1.0000	T			
153	0.0	T			
165	0.0	T			
166	0.0	T			
167	0.0	T			
168	0.0	T			
169	1.0000	T			
170	0.0	T			
171	0.0	T			
172	0.0	T			
173	0.0	T			
174	0.0	T			
175	0.0	T			
176	0.0	T			
177	1.0000	T			

FIG. 8.21 COMPUTER OUTPUT FOR EXAMPLE 3



THE FOLLOWING NUMERICAL DATA HAS BEEN SUPPLIED FOR CYCLE NUMBER 2

KGLOB	DATAK
65	10.3800
59	5.0600
58	0.7480
64	0.4570
63	8.8800
79	82.8000
80	73.6000
46	19.4000
48	7.5700
188	2.5800
208	4.4400
189	2.8000
138	3646.2700
190	853.5000
440	25.5000
441	38.8000

DATA PRINTED AGAIN FOR CHECKING. ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE

KGLOB	DATAK	PRD
1	1.0000	T
2	0.0	T
10	1.0000	T
11	0.0	T
12	0.0	T
13	0.0	T
14	0.0	T
15	0.0	T
16	0.0	T
17	0.0	T
23	1.0000	T
24	0.0	T
25	0.0	T
26	1.0000	T
27	0.0	T
28	1.0000	T
29	0.0	T
30	0.0	T
40	1.0000	T
41	0.0	T
46	19.4000	T
48	7.5700	T
58	0.7480	T
59	5.0600	T
63	8.8800	T
64	0.4570	T
65	10.3800	T
76	44.0000	T
78	29000.0000	T
79	82.8000	T
80	73.6000	T
95	0.0	T
96	1.0000	T
101	0.0	T
119	1.0000	T
120	129.6000	T
121	0.0	T
122	0.0	T
123	0.0	T
124	1.0000	T
125	1.0000	T
126	0.0	T
127	1.0000	T
128	1.0000	T
129	1.0000	T

FIG. 8.22 COMPUTER OUTPUT FOR EXAMPLE 3





130	1.0000	T
132	1440.4500	T
133	400.9500	T
138	3646.2700	T
153	0.0	T
165	0.0	T
166	0.0	T
167	0.0	T
168	0.0	T
169	1.0000	T
170	0.0	T
171	0.0	T
172	0.0	T
173	0.0	T
174	0.0	T
175	0.0	T
176	0.0	T
177	1.0000	T
178	1.0000	T
180	0.0	T
185	1.0000	T
186	114.0000	T
187	114.0000	T
188	2.5800	T
189	2.8000	T
190	853.5000	T
194	1.0000	T
197	0.9000	T
208	4.4400	T
214	1.0000	T
216	114.0000	T
220	1.0000	T
221	1.0000	T
222	0.0	T
223	0.0	T
224	0.0	T
225	0.0	T
226	1.0000	T
290	1.0000	T
291	0.0	T
310	0.8500	T
320	1.0000	T
321	0.0	T
322	0.0	T
323	0.0	T
324	1.0000	T
342	1.0000	T
343	0.0	T
344	0.0	T
400	0.9000	T
401	1.2500	T
402	1.5000	T
403	1.5000	T
404	1.2500	T
405	1.0000	T
406	90.0000	T
407	21.0000	T
408	0.0	T
409	0.0	T
410	0.0	T
411	1067.0000	T
412	0.0	T
413	0.0	T
414	0.0	T
415	297.0000	T
416	0.0	T
417	0.0	T
440	25.5000	T
441	38.8000	T

FIG. 8.23 COMPUTER OUTPUT FOR EXAMPLE 3



```

CYCLE NUMBER 2      ***  START EXECUTION WITH TABLE 1  ***

  SCANNING OF TABLE 1 IS COMPLETE. RULE NO. 1 APPLIES

SUSPENDED EXECUTION OF TABLE 1 AT ACTION 1 OF RULE 1
STARTED EXECUTION OF TABLE 2 FOR DIRECT EXECUTION

  SCANNING OF TABLE 2 IS COMPLETE. RULE NO. 1 APPLIES

SUSPENDED EXECUTION OF TABLE 2 AT ACTION 1 OF RULE 1
STARTED EXECUTION OF TABLE 3 FOR DIRECT EXECUTION

  SCANNING OF TABLE 3 IS COMPLETE. RULE NO. 13 APPLIES

SUSPENDED EXECUTION OF TABLE 3 AT ACTION 6 OF RULE 13
STARTED EXECUTION OF TABLE 56 FOR DIRECT EXECUTION

  SCANNING OF TABLE 56 IS COMPLETE. RULE NO. 1 APPLIES

SUSPENDED EXECUTION OF TABLE 56 AT ACTION 1 OF RULE 1
STARTED EXECUTION OF TABLE 57 FOR DIRECT EXECUTION

SUSPENDED EXECUTION OF TABLE 57 AT CONDITION 5 OF RULE 2
STARTED EXECUTION OF TABLE 5 TO OBTAIN VALUE OF DATA NUMBER 97

SUSPENDED EXECUTION OF TABLE 5 AT CONDITION 7 OF RULE 7
STARTED EXECUTION OF TABLE 6 TO OBTAIN VALUE OF DATA NUMBER 99

  SCANNING OF TABLE 6 IS COMPLETE. RULE NO. 2 APPLIES

SUSPENDED EXECUTION OF TABLE 6 AT ACTION 1 OF RULE 2
STARTED EXECUTION OF TABLE 14 FOR DIRECT EXECUTION

  SCANNING OF TABLE 14 IS COMPLETE. RULE NO. 3 APPLIES

SUSPENDED EXECUTION OF TABLE 14 AT ACTION 3 OF RULE 3
STARTED EXECUTION OF TABLE 17 FOR DIRECT EXECUTION

  SCANNING OF TABLE 17 IS COMPLETE. RULE NO. 1 APPLIES

SUSPENDED EXECUTION OF TABLE 17 AT ACTION 1 OF RULE 1
STARTED EXECUTION OF TABLE 18 FOR DIRECT EXECUTION

  SCANNING OF TABLE 18 IS COMPLETE. RULE NO. 1 APPLIES

  RESTART EXECUTION OF TABLE 17 AT ACTION 1 OF RULE 1

  RESTART EXECUTION OF TABLE 14 AT ACTION 3 OF RULE 3

  RESTART EXECUTION OF TABLE 6 AT ACTION 1 OF RULE 2

  RESTART EXECUTION OF TABLE 5 AT CONDITION 7 OF RULE 7

SUSPENDED EXECUTION OF TABLE 5 AT CONDITION 11 OF RULE 7
STARTED EXECUTION OF TABLE 20 TO OBTAIN VALUE OF DATA NUMBER 110

  SCANNING OF TABLE 20 IS COMPLETE. RULE NO. 2 APPLIES

SUSPENDED EXECUTION OF TABLE 20 AT ACTION 1 OF RULE 2
STARTED EXECUTION OF TABLE 21 FOR DIRECT EXECUTION

  SCANNING OF TABLE 21 IS COMPLETE. RULE NO. 5 APPLIES

  RESTART EXECUTION OF TABLE 20 AT ACTION 1 OF RULE 2

  RESTART EXECUTION OF TABLE 5 AT CONDITION 11 OF RULE 7

  SCANNING OF TABLE 5 IS COMPLETE. RULE NO. 7 APPLIES

SUSPENDED EXECUTION OF TABLE 5 AT ACTION 7 OF RULE 7
STARTED EXECUTION OF TABLE 7 FOR DIRECT EXECUTION

  SCANNING OF TABLE 7 IS COMPLETE. RULE NO. 7 APPLIES

  RESTART EXECUTION OF TABLE 5 AT ACTION 7 OF RULE 7

  RESTART EXECUTION OF TABLE 57 AT CONDITION 5 OF RULE 2

  SCANNING OF TABLE 57 IS COMPLETE. RULE NO. 2 APPLIES

```

FIG. 8.24 COMPUTER OUTPUT FOR EXAMPLE 3



STARTED EXECUTION OF TABLE 65  
 SUSPENDED EXECUTION OF TABLE 65 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 313  
 STARTED EXECUTION OF TABLE 66  
 SCANNING OF TABLE 66 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 66 AT ACTION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 215  
 STARTED EXECUTION OF TABLE 4  
 SCANNING OF TABLE 4 IS COMPLETE. RULE NO. 4 APPLIES  
 RESTART EXECUTION OF TABLE 66 AT ACTION 1 OF RULE 1  
 RESTART EXECUTION OF TABLE 65 AT CONDITION 1 OF RULE 1  
 SCANNING OF TABLE 65 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 2  
 SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 2  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 299  
 STARTED EXECUTION OF TABLE 62  
 SCANNING OF TABLE 62 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 62 AT ACTION 1 OF RULE 1  
 STARTED EXECUTION OF TABLE 63 FOR DIRECT EXECUTION  
 SUSPENDED EXECUTION OF TABLE 63 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 137  
 STARTED EXECUTION OF TABLE 51  
 SCANNING OF TABLE 51 IS COMPLETE. RULE NO. 1 APPLIES  
 SUSPENDED EXECUTION OF TABLE 51 AT ACTION 2 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 195  
 STARTED EXECUTION OF TABLE 55  
 SCANNING OF TABLE 55 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 51 AT ACTION 2 OF RULE 1  
 RESTART EXECUTION OF TABLE 63 AT CONDITION 1 OF RULE 1  
 SCANNING OF TABLE 63 IS COMPLETE. RULE NO. 1 APPLIES  
 RESTART EXECUTION OF TABLE 62 AT ACTION 1 OF RULE 1

FIG. 8.25 COMPUTER OUTPUT FOR EXAMPLE 3



SUSPENDED EXECUTION OF TABLE 57 AT ACTION 1 OF RULE 2  
 STARTED EXECUTION OF TABLE 58 FOR DIRECT EXECUTION

SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 1 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 295  
 STARTED EXECUTION OF TABLE 60

SCANNING OF TABLE 60 IS COMPLETE. RULE NO. 1 APPLIES

RESTART EXECUTION OF TABLE 58 AT CONDITION 1 OF RULE 1

SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 2 OF RULE 1  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 298  
 STARTED EXECUTION OF TABLE 67

SCANNING OF TABLE 67 IS COMPLETE. RULE NO. 1 APPLIES

RESTART EXECUTION OF TABLE 58 AT CONDITION 2 OF RULE 1

SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 2  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 297

RESTART EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 2

SUSPENDED EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 2  
 REASON: MISSING INGREDIENT CORRESPONDING TO DATA NUMBER 311  
 STARTED EXECUTION OF TABLE 68

SCANNING OF TABLE 68 IS COMPLETE. RULE NO. 1 APPLIES

RESTART EXECUTION OF TABLE 58 AT CONDITION 3 OF RULE 2

SCANNING OF TABLE 58 IS COMPLETE. RULE NO. 3 APPLIES

\*\*\*\*\* STRENGTH AND STABILITY CRITERIA SATISFIED.\*\*\*\*\*

RESTART EXECUTION OF TABLE 57 AT ACTION 1 OF RULE 2

RESTART EXECUTION OF TABLE 56 AT ACTION 1 OF RULE 1

RESTART EXECUTION OF TABLE 3 AT ACTION 6 OF RULE 13

RESTART EXECUTION OF TABLE 2 AT ACTION 1 OF RULE 1

RESTART EXECUTION OF TABLE 1 AT ACTION 1 OF RULE 1

FIG. 8.26 COMPUTER OUTPUT FOR EXAMPLE 3





DATA VALUES AT THE END OF CYCLE NO. 2  
ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE

KGLOB	DATAK	PRD			
			160	0.0	T
			185	1.0000	T
			186	114.0000	T
			187	114.0000	T
			188	2.5800	T
			189	2.8000	T
			190	853.5000	T
			194	1.0000	T
			195	127.9451	T
			196	150.8156	T
			197	0.9000	T
			208	4.4400	T
			214	1.0000	T
			215	44.1860	T
			216	114.0000	T
			217	44.1860	T
			219	25.6757	T
			220	1.0000	T
			221	1.0000	T
			222	0.0	T
			223	0.0	T
			224	0.0	T
			225	0.0	T
			226	1.0000	T
			254	1.0000	T
			255	0.0	T
			256	0.0	T
			257	0.0	T
			266	1.0000	T
			267	0.0	T
			268	0.0	T
			275	0.0	T
			290	1.0000	T
			291	0.0	T
			295	3281.6428	T
			296	1536.4797	T
			297	658.7854	T
			298	768.2395	T
			299	3281.6428	T
			310	0.8500	T
			311	8416.3516	T
			312	2841.8264	T
			313	0.5481	T
			314	1.0000	T
			315	0.0	T
			316	0.0	T
			317	0.0	T
			320	1.0000	T
			321	0.0	T
			322	0.0	T
			323	0.0	T
			324	1.0000	T
			335	1.0000	T
			336	1.0000	T
			337	1.0000	T
			342	1.0000	T
			343	0.0	T
			344	0.0	T
			400	0.9000	T
			401	1.2500	T
			402	1.5000	T
			403	1.5000	T
			404	1.2500	T
			405	1.0000	T
			406	90.0000	T
			407	21.0000	T
			408	0.0	T
			409	0.0	T
			410	0.0	T
			411	1067.0000	T
			412	0.0	T
			413	0.0	T
			414	0.0	T
			415	297.0000	T
			416	0.0	T
			417	0.0	T
			440	25.5000	T
			441	38.8000	T
			EXECUTION OF PROGRAM IS COMPLETED		
			\$\$SIGNOFF		
1	1.0000	T			
2	0.0	T			
10	1.0000	T			
11	0.0	T			
12	0.0	T			
13	0.0	T			
14	0.0	T			
15	0.0	T			
16	0.0	T			
17	0.0	T			
23	1.0000	T			
24	0.0	T			
25	0.0	T			
26	1.0000	T			
27	0.0	T			
28	1.0000	T			
29	0.0	T			
30	0.0	T			
40	1.0000	T			
41	0.0	T			
46	19.4000	T			
48	7.5700	T			
58	0.7480	T			
59	5.0600	T			
63	8.8800	T			
64	0.4570	T			
65	10.3800	T			
76	44.0000	T			
78	29000.0000	T			
79	82.8000	T			
80	73.6000	T			
95	1.0000	T			
96	0.0	T			
97	0.0	T			
98	0.0	T			
99	1.0000	T			
100	0.0	T			
101	0.0	T			
102	0.0	T			
110	1.0000	T			
111	0.0	T			
112	0.0	T			
113	0.0	T			
119	1.0000	T			
120	129.6000	T			
121	0.0	T			
122	0.0	T			
123	0.0	T			
124	1.0000	T			
125	1.0000	T			
126	0.0	T			
127	1.0000	T			
128	1.0000	T			
129	1.0000	T			
130	1.0000	T			
132	1440.4500	T			
133	400.9500	T			
136	1.0000	T			
137	14556.3047	T			
138	3646.2700	T			
139	1.0000	T			
153	0.0	T			
165	0.0	T			
166	0.0	T			
167	0.0	T			
168	0.0	T			
169	1.0000	T			
170	0.0	T			
171	0.0	T			
172	0.0	T			
173	0.0	T			
174	0.0	T			
175	0.0	T			
176	0.0	T			
177	1.0000	T			
178	1.0000	T			

FIG. 8.27 COMPUTER OUTPUT FOR EXAMPLE 3



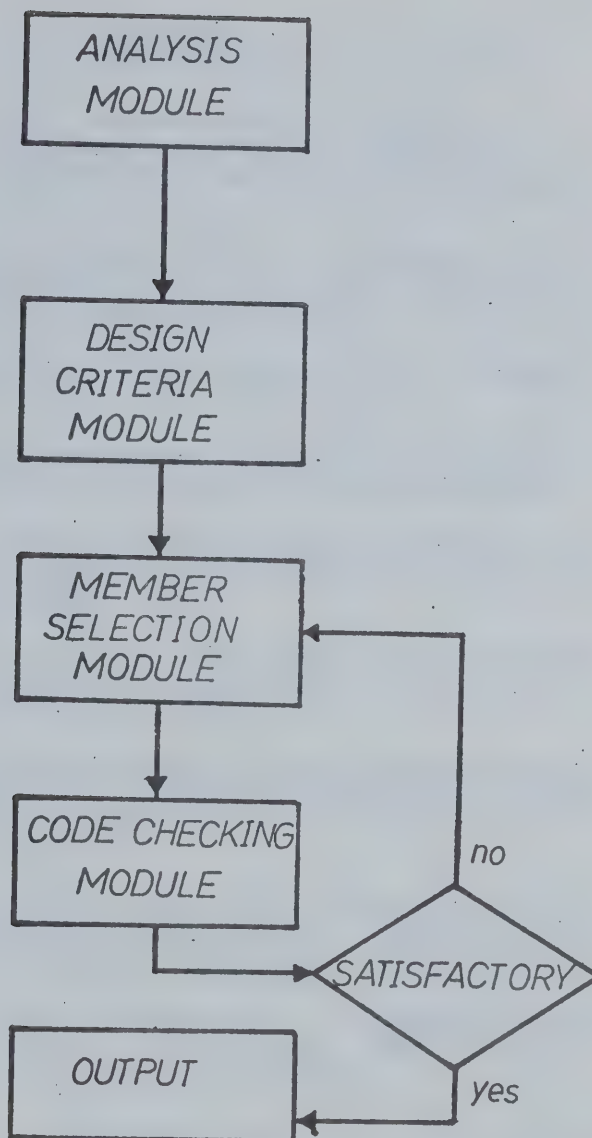


FIG. 9.1 THE COMPLETE DESIGN PROCESS



## REFERENCES

1. ACI Committee 118. Decision Logic Tables For Building Code Requirements for Reinforced Concrete (ACI 318-71), ACI Journal, December 1973.
2. Allen, D.E., Limit States Design: A Probabilistic Study. Canadian Journal of Civil Engineering, Vol. 2, March, 1975, pp. 36-49.
3. Canadian Institute of Steel Construction. Column Selection Program (2), User's Manual.
4. Canadian Institute of Steel Construction, Floor System Selection Program (2), User's Manual.
5. Carlson, W.M., Engineering Approach to Information Processing, Journal of the Structural Division, ASCE, ST4, August, 1963.
6. Fenves, S.J., Tabular Decision Logic for Structural Design, Journal of the Structural Division, ASCE, ST6, December, 1966.
7. Fenves, S.J., Representation of the Computer-Aided Design Process by a Network of Decision Tables. Computer and Structures, Vol. 3, No. 5, September, 1973.
8. Fenves, S.J., Scenario for a Third Computer Revolution in Structural Engineering. Journal of the Structural Division, ASCE, ST1, January, 1971.
9. Fenves, S.J., Needs and Prospects for Computer-Aided Structural Design, Proceedings, 1966 Illinois Structural Engineering Conference.
10. Gilersleeve, T.R., Decision Tables and Their Practical Application in Data Processing, Prentice-Hall Inc., 1970.
11. Gilmor, M.I. and Selby, K.A., Structural Steel Design Using STRUDL, Canadian Structural Engineering Conference, 1970.
12. Goel, S.K., Computer-Aided Processing of Structural Design Specifications, Ph.D. Thesis, University of Illinois, 1970.



13. Hatfield, F.J. and Fenves, S.J., The Information Organizer: A System for Symbolic Data Manipulation, Computer and Structures, Vol. 1, 1973.
14. Hughes, M.L., Shank, R.M. and Stein, E.S., Decision Tables, Management Development Institute Publications, Division of Information Industries Inc., Wayne, Penn. U.S.A.
15. Krentz, H.A., CSA Standard S16-1969, Steel Structures for Buildings, Canadian Structural Engineering Conference, 1970.
16. Logcher, R.D. and Sturman, G.M., STRUDL - A Computer System for Structural Design, Journal of the Structural Division, ASCE, ST6, December, 1966.
17. Logcher, R.D., Mozzotta, S.G. and Teague, L.C., Languages for User Defined Member Design Processes, Journal of the Structural Division, ASCE, June, 1967.
18. London, K.R., Decision Tables, Auerbach Publishers, 1972.
19. Lopez, L.A., FILES: Automated Engineering Data Management System, Journal of the Structural Division, ASCE, April, 1975.
20. McDaniel, H., Introduction to Decision Logic Tables, J. Wiley and Sons, 1968.
21. Miller, C.L., Man-Machine Communications in Civil Engineering, Journal of the Structural Division, ASCE, ST4, August, 1963.
22. Noland, J. and Feng, C.C., Formulation of Decision Logic Tables, Journal of the Structural Division ASCE, ST1, January, 1971.
23. Noland, J. and Feng, C.C., ACI Building Code in Decision Table Format, Journal of the Structural Division, ASCE, ST4, April, 1975.
24. Nyman, D.J. and Fenves, S.J., Organizational Model for Design Specifications, Journal of the Structural Division, ASCE, ST4, April, 1975.





25. Pollack, S.L., Hicks, H.T. and Harrison, W.J.,  
Decision Tables: Theory and Practice, J.  
Wiley, 1971.
26. Shaw, C., Decision Tables, System Development Corp.,  
Santa Monica, California 1965.
27. Springfield, J., Application of Computers in Stru-  
ctural Engineering Design, Canadian Structural  
Engineering Conference, 1970.
28. Vahl, T., Present and Future Possibilities Within  
Computer Aided Design, Computer and Structures,  
Vol. 4, 1974.
29. Wright, R.M., Boyle, L.T. and Melin, J.W., Constraint  
Processing in Design, Journal of the Structural  
Division, ASCE, ST1, January, 1971.



APPENDIX A  
DECISION TABLE HEIRACHY CHARTS  
AND  
DECISION TABLE INDEX

Figs. A.1 to A.11 present the heirachy charts of the decision tables in Appendix B. These charts illustrate the order of execution of decision tables when a particular task is performed. The full lines in the charts represent direct execution and the broken lines represent conditional execution.

Fig. A.12 provides an explanation of the decision table designation used in Figs. A.1 to A.11.

A decision table index is provided in Fig. A.13 where the table number for computer input is tabulated with the corresponding code designation. In referring to a decison table in the text, both systems of designation are used. For example, Decison Table X.2 of Fig. A.1 is referred to as X.2(2) where the number in brackets indicates the computer input number for this table.



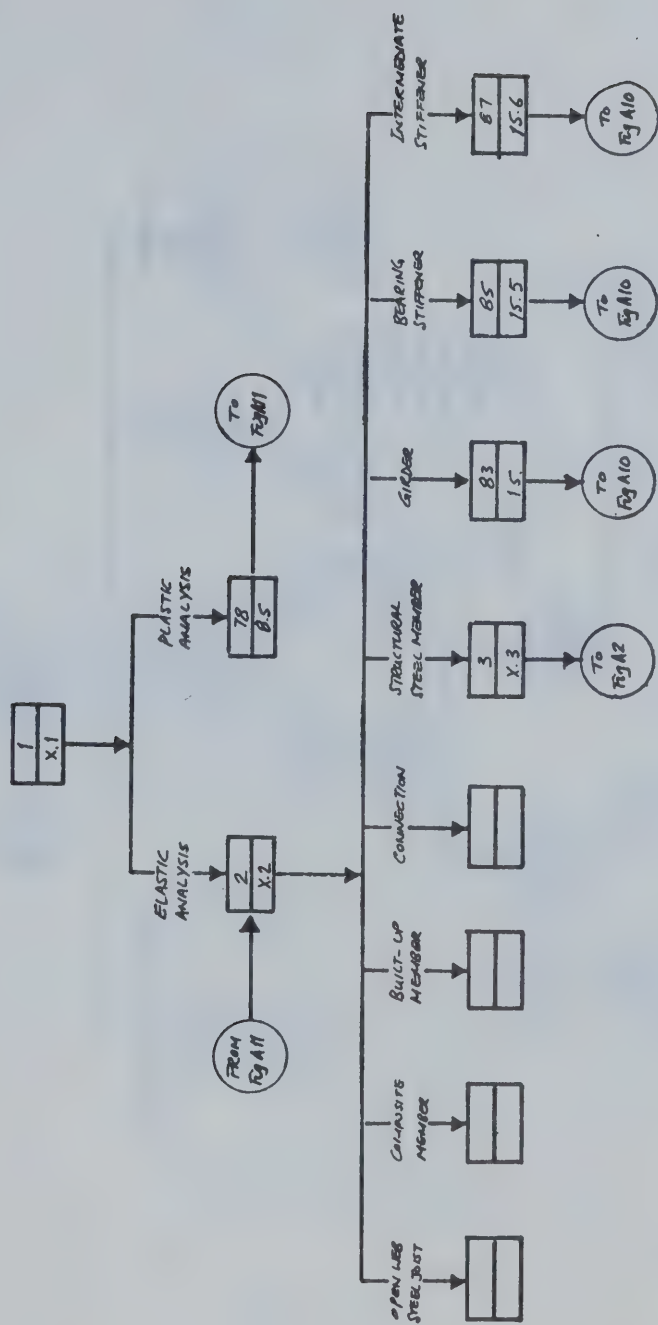


FIG. A.1 DECISION TABLE HEIRACHY CHART 1



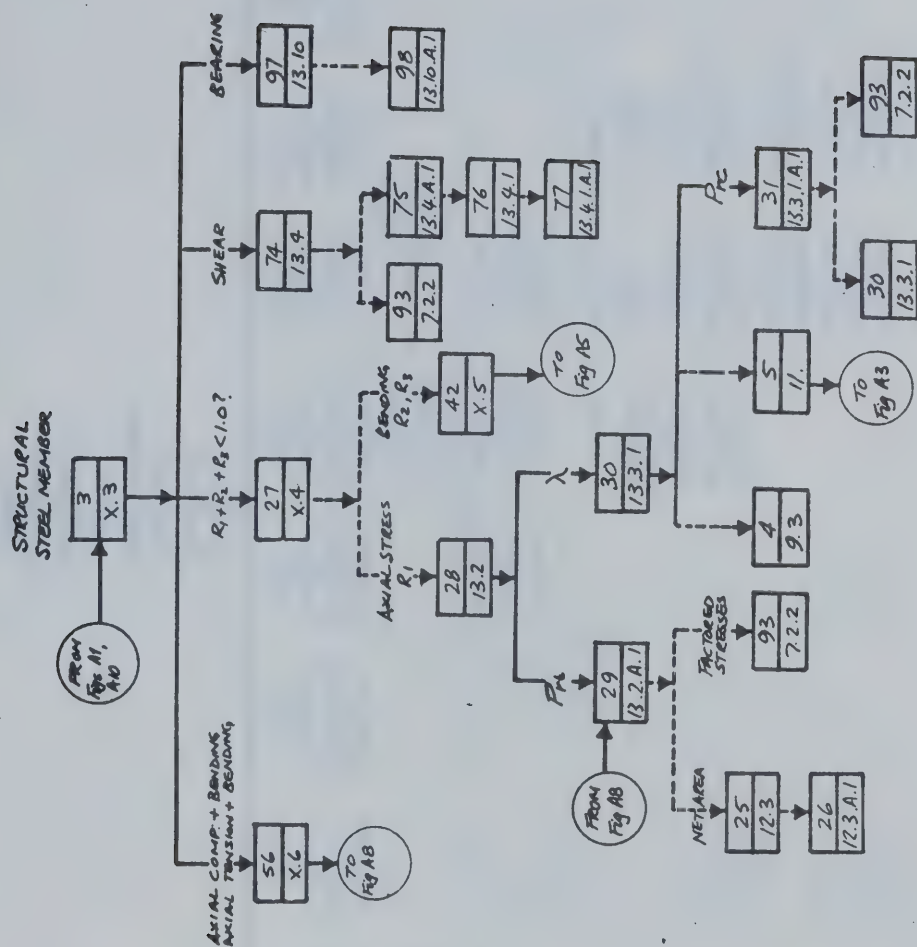


FIG. A.2 DECISION TABLE HEIRACHY CHART 2





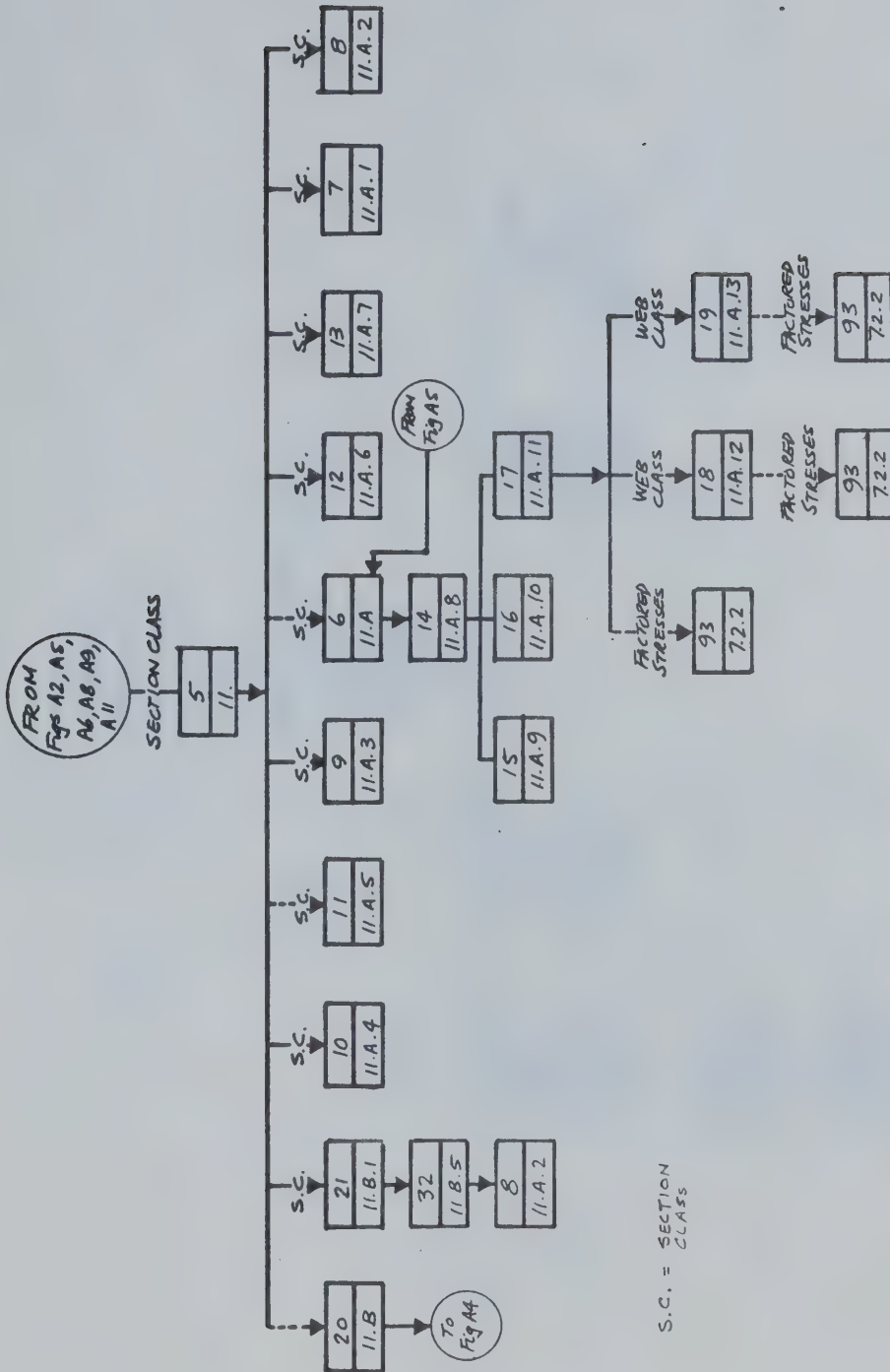


FIG. A.3 DECISION TABLE HEIRACHY CHART 3



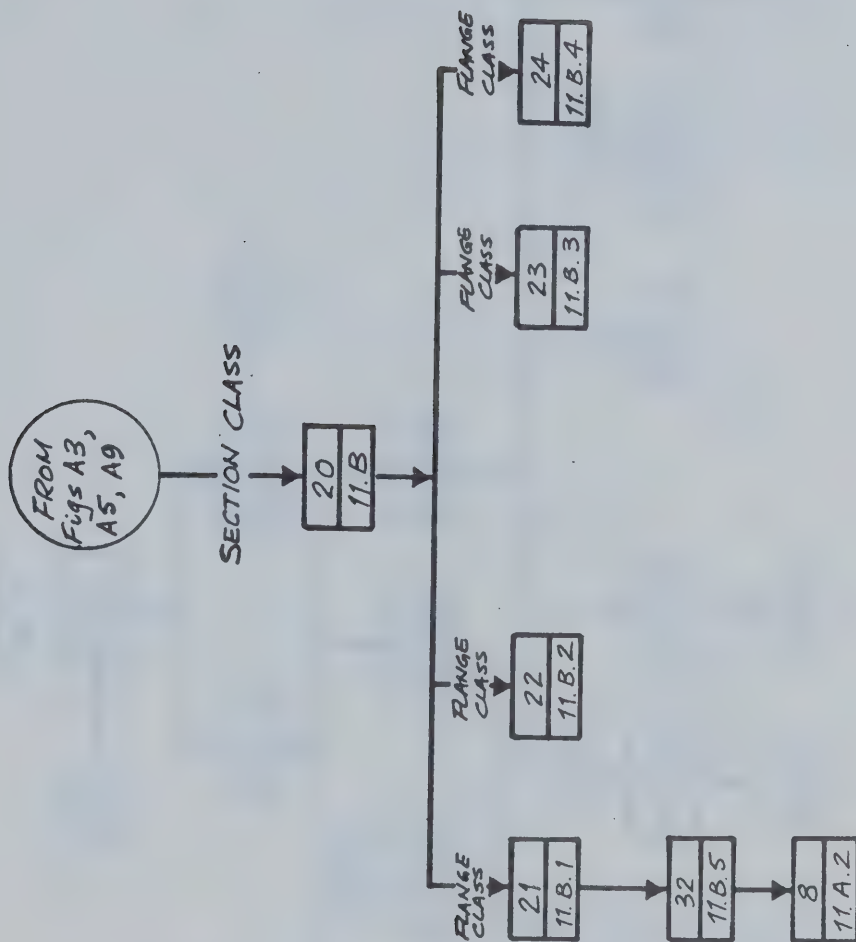


FIG. A.4 DECISION TABLE HEIRACHY CHART 4



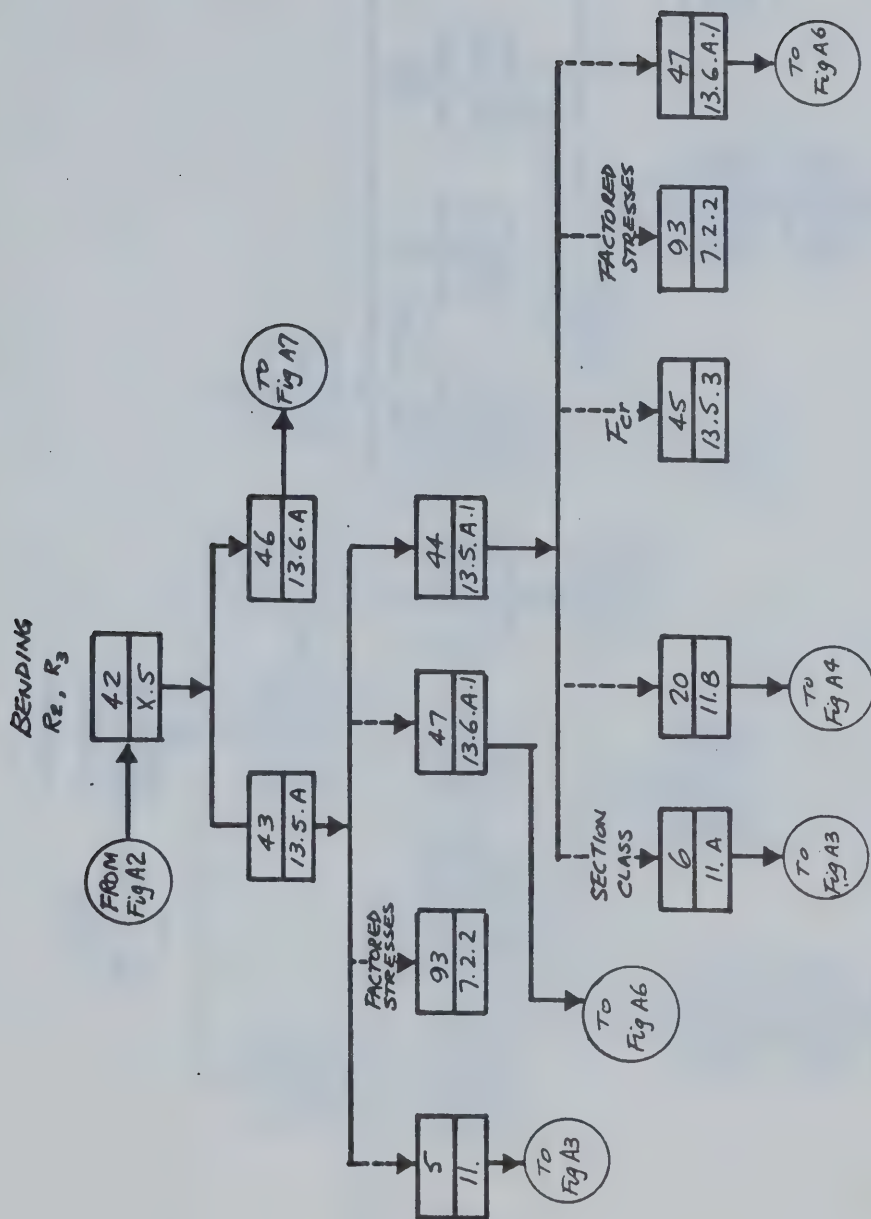


FIG. A.5 DECISION TABLE HEIRACHY CHART 5









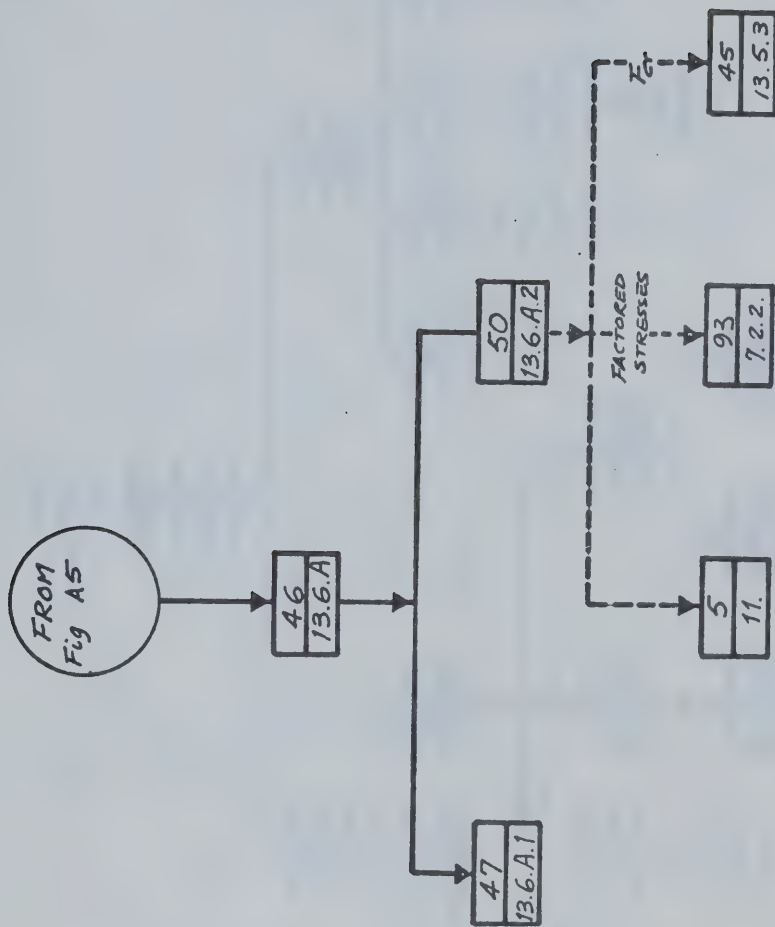


FIG. A.7 DECISION TABLE HEIRACHY CHART 7



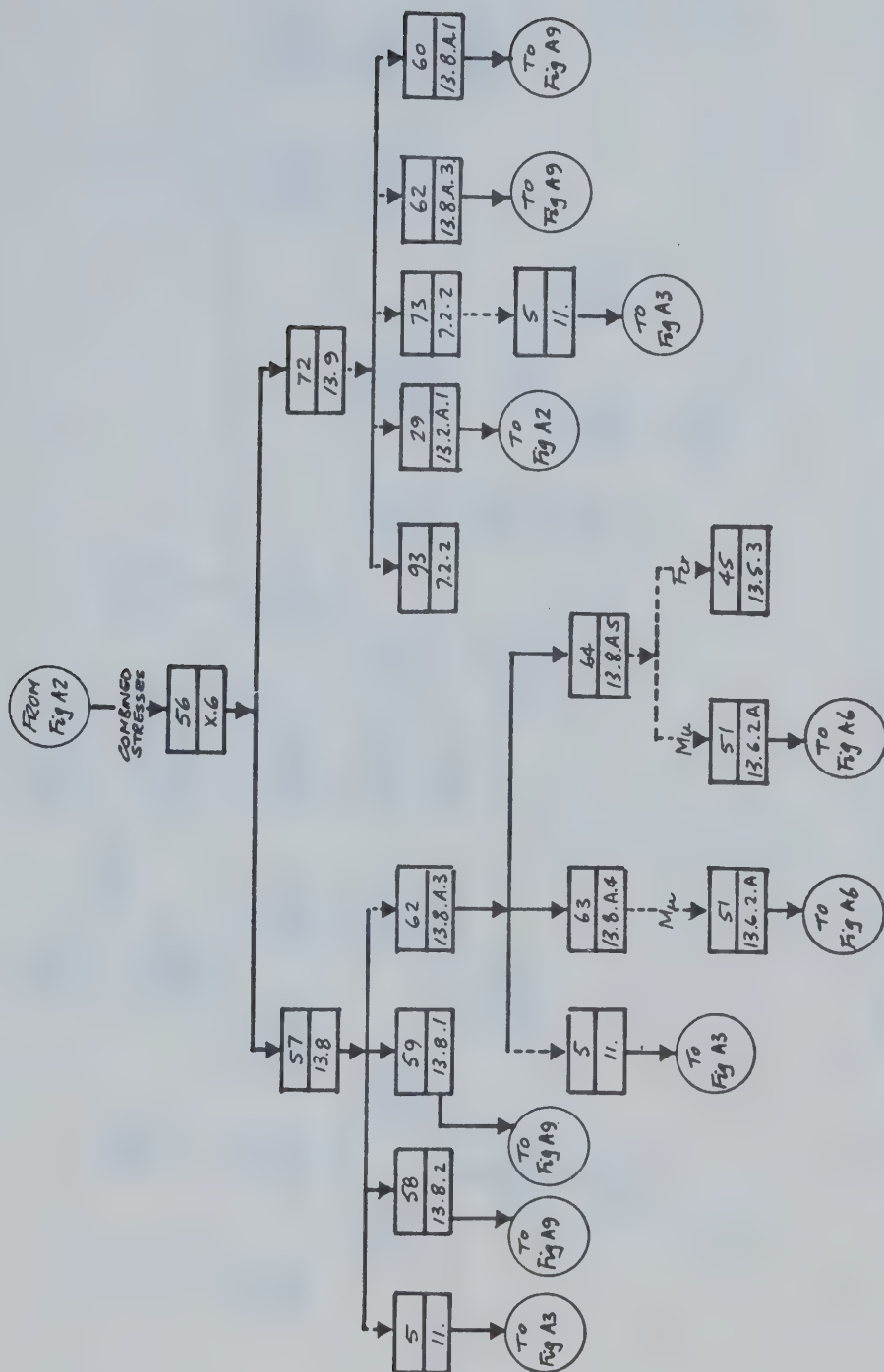


FIG. A.8 DECISION TABLE HEIRACHY CHART 8



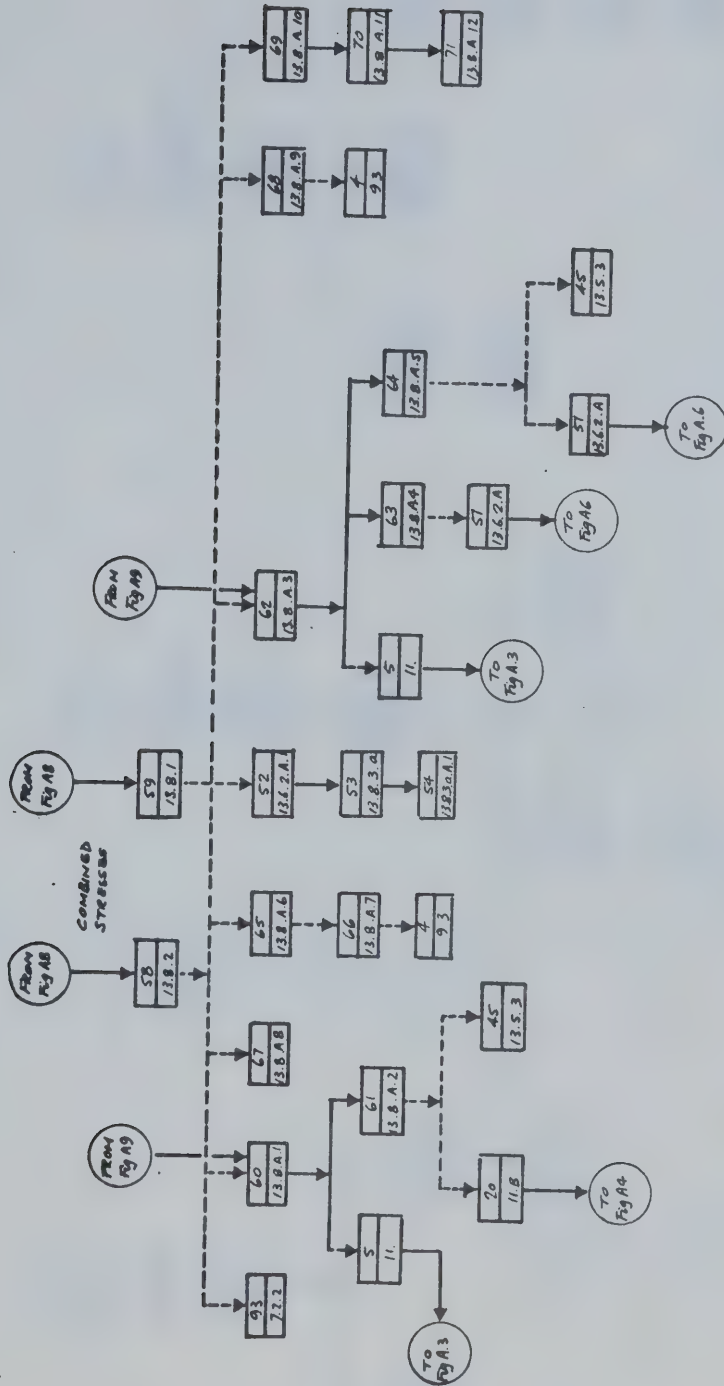


FIG. A.9 DECISION TABLE HEIRACHY CHART 9



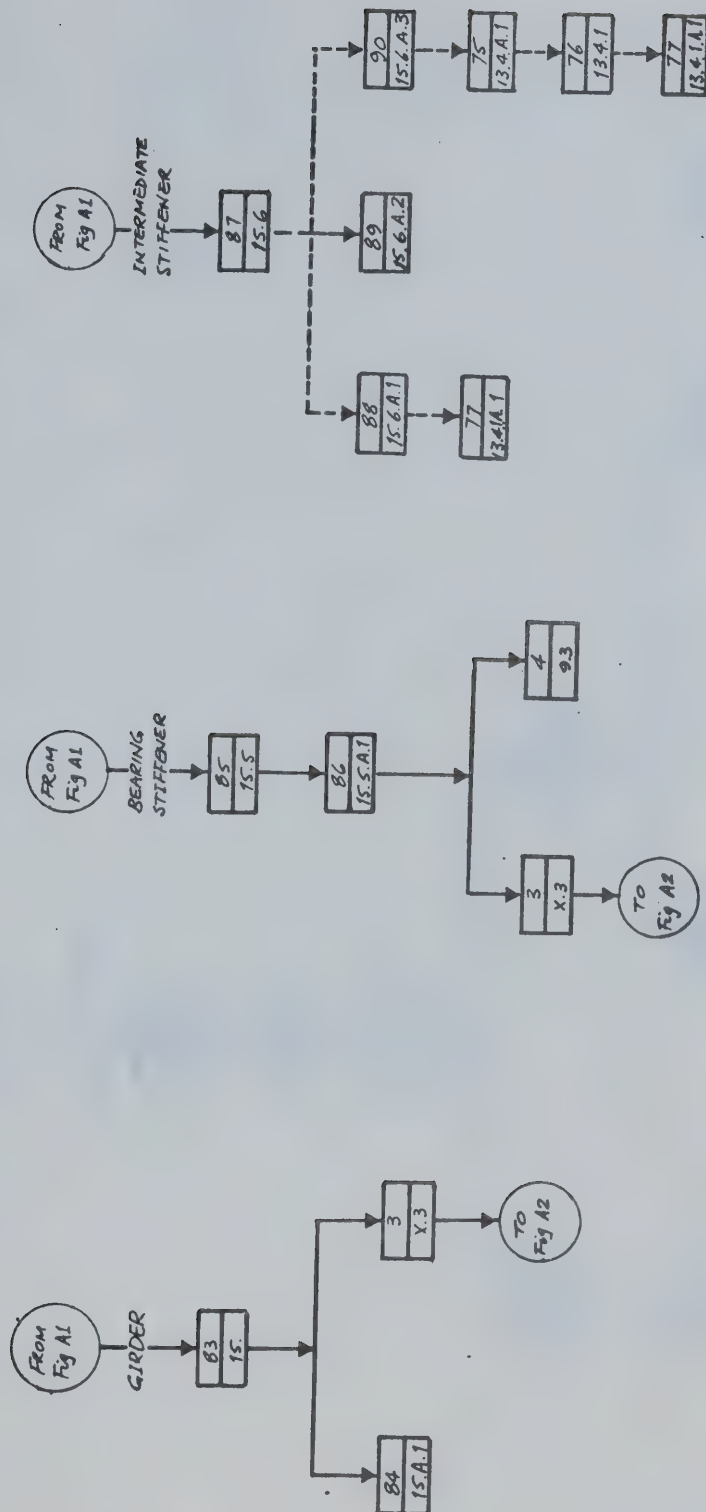


FIG. A.10 DECISION TABLE HIERARCHY CHART 10





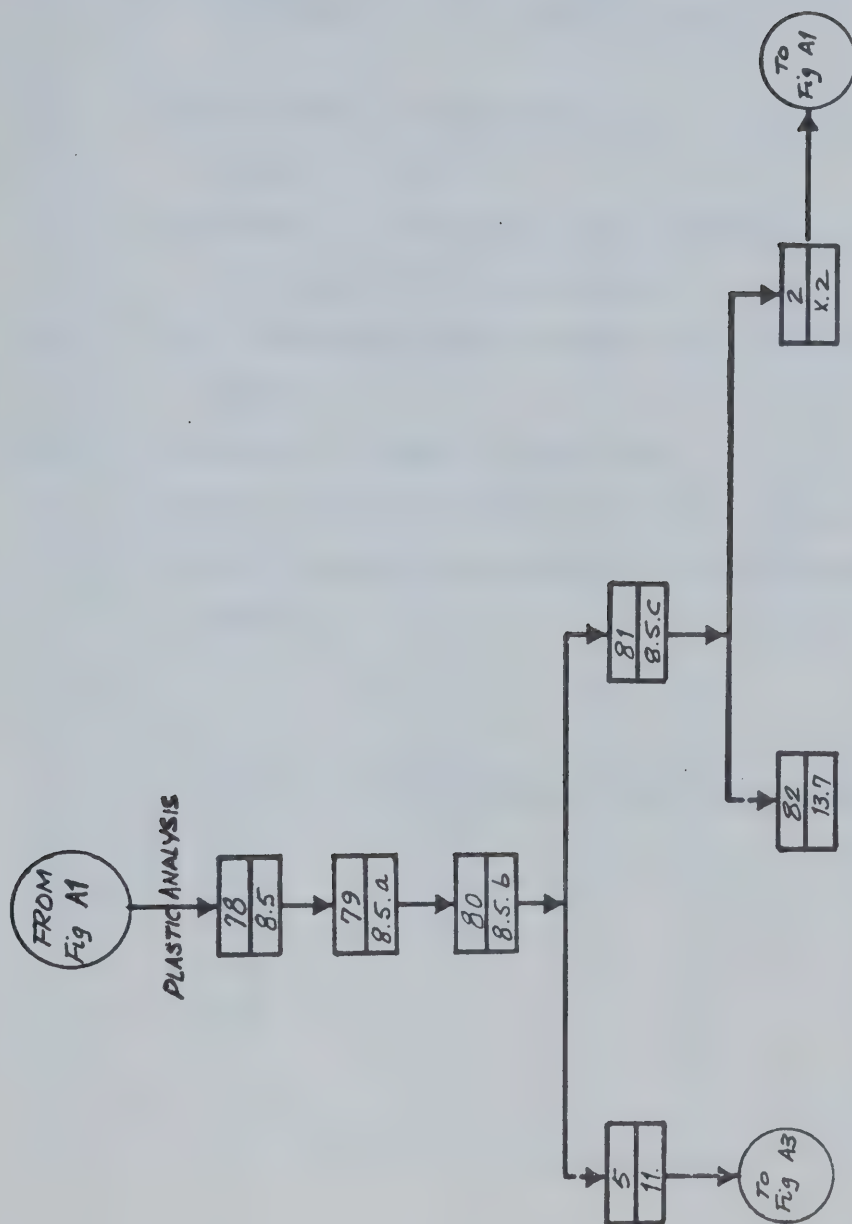


FIG. A.11 DECISION TABLE HEIRACHY CHART 11



T
X. m

T
Sect. $\theta$ . n

- T = Table Number of Computer Input. (T is an integer number)
- X = Indicates a "Switching Table" which does not correspond to a Code section.
- m = The "Switching Table" number (m is an integer number)
- Sect = Section or clause of the Code.
- $\theta$  = Level 1 subtable designation ( $\theta$  is either A or B)
- n = Level 2 subtable designation (n is an integer number).

Fig. A.12 Explanation of Table Designation



S16.1 SECTION NUMBER	DECISION TABLE NUMBER	S16.1 SECTION NUMBER	DECISION TABLE NUMBER	S16.1 SECTION NUMBER	DECISION TABLE NUMBER
X.1	1		39	13.4.1.A.1	77
X.2	2		40	8.5	78
X.3	3		41	8.5.a.	79
9.3	4	X.5	42	8.5.b.	80
11.	5	13.5.A	43	8.5.c.	81
11.A.	6	13.5.A.1	44	13.7	82
11.A.1	7	13.5.3	45	15.	83
11.A.2	8	13.6.A	46	15.A.1	84
11.A.3	9	13.6.A.1	47	15.5	85
11.A.4	10	13.6.1	48	15.5.A.1	86
11.A.5	11	13.6.2	49	15.6	87
11.A.6	12	13.6.A.2	50	15.6.A.1	88
11.A.7	13	13.6.2.A	51	15.6.A.2	89
11.A.8	14	13.6.2.A.1	52	15.6.A.3	90
11.A.9	15	13.8.3.a	53	13.9	91
11.A.10	16	13.8.3.a.A.1	54	13.9.A.1	92
11.A.11	17	13.6.1.A.1	55	7.2.2	93
11.A.12	18	X.6	56	7.2.3	94
11.A.13	19	13.8	57	7.2.4	95
11.B	20	13.8.2	58	7.2.5	96
11.B.1	21	13.8.1	59	13.10	97
11.B.2	22	13.8.A.1	60	13.10.A.1	98
11.B.3	23	13.8.A.2	61		99
11.B.4	24	13.8.A.3	62		100
12.3	25	13.8.A.4	63		101
12.3.A.1	26	13.8.A.5	64		102
X.4	27	13.8.A.6	65		103
13.2	28	13.8.A.7	66		104
13.2.A.1	29	13.8.A.8	67		105
13.3.1	30	13.8.A.9	68		106
13.3.1.A.1	31	13.8.A.10	69		107
11.B.5	32	13.8.A.11	70		108
	33	13.8.A.12	71		109
	34	13.9	72		110
	35	13.9.A.1	73		111
	36	13.4	74		112
	37	13.4.A.1	75		113
	38	13.4.1	76		114

FIG. A13 Decision Table Index



APPENDIX B

GLOBAL DATA SHEETS, DATA DEPENDENT LISTS  
AND  
DECISION TABLES OF CSA S16.1





DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
1	\$ELAN		1	Elastic Analysis		*
2	\$PLAN		1	Plastic Analysis		*
10	\$SSM		2	Structural Steel Member		
11	\$OWSJ		2	Open-Web Steel Joists		
12	\$COM		2	Composite Construction Member		
13	\$BUM		2	Built-Up Member		
14	\$CON		2	Connection		
15	\$GIRD		2	Girder		
16	\$BSTI		2	Bearing Stiffener		
17	\$INST		2	Intermediate Transverse Stiffener		
19	\$TEN			Tension Member Without Holes		*
20	\$MPCON			Member Using Pin-Connections		*
21	\$LSM		4	Continuously Laterally Supported Member		*
22	\$LUSM		4	Laterally Unsupported Member		*
23	\$CPM		6	Compression Member		*
24	\$TENM		6	Tension Member		*
25	\$CTRUS			Compression Member In Truss		*
26	\$BELS		8	Beam With Both Ends With Effective Lateral Support To The Compression Flange		*
27	\$BOELS		8	Beam With Only One End With Effective Lateral Support For The Compression Flange		*
28	\$LOAD1		10	Member Not Subjected To Transverse Loads In The Minor Axis Plane Between Supports		*
29	\$LOAD2		10	Member Subjected To Distributed Load Or Series Of Point Loads In The Minor Axis Plane Between Supports		*
30	\$LOAD3		10	Member Subjected To A Concentrated Load Or Moment In The Minor Axis Plane Between Supports		*
40	\$CURV1		12	Member Bent In Single Curvature About The Major Axis		*
41	\$CURV2		12	Member Bent In Double Curvature About The Major Axis		*
45	\$ANET	12.3		Net Area	A <sub>n</sub>	*
46	\$AGRS			Gross Area/Nominal Area of Pin	A <sub>g</sub>	*
47	\$AW			Web Area	A <sub>w</sub>	*
48	\$AF			Flange Area	A <sub>f</sub>	*
55	\$W			Width of Element (In A Plate)		*

\*See lists of dependents



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
56	W1			Width Of One Angle Leg	t	*
57	W2			Width Of The Other Leg Of Angle	b	*
58	ST			Flange Thickness	d	*
59	S8			Flange Width	n	*
60	D1			Outside Dia. Of Circular Section	h	*
61	D2			Dia. of Rivet Or Bolt/Overall Depth Of A Section	w	*
62	FNU			Number Of Holes In A Tension Member	d	*
63	SH			Depth Of Web	t	*
64	SW			Thickness Of Web	t <sub>web</sub>	*
65	SD			Depth of Section	F <sub>cr</sub>	*
66	TP			Thickness Of Plate	F <sub>y</sub>	*
67	TWEB			Thickness Of Web Of Girder	F <sub>u</sub>	*
75	FCR	13.5.3		Critical Plate Buckling Stress	E	*
76	FY			Specified Minimum Yield Stress, Or Yield Strength	z	*
77	FU			Specified Minimum Tensile Strength	s	*
78	E			Elastic Modulus Of Steel (29000 ksi Assumed)	G	*
79	z			Elastic Modulus Of A Steel Section	J	*
80	S			Elastic Modulus Of A Steel Section	C <sub>w</sub>	*
81	G			Shear Modulus Of Steel (11200 ksi Assumed)		
82	VJ			St. Venant's Torsional Constant		
83	CW			Warping Torsional Constant		
95	\$CLAS1	11	14	Set Of Section = 1		*
96	\$CLAS2	11	14	Class Of Section = 2		*
97	\$CLAS3	11	14	Class Of Section = 3		*
98	\$CLAS4	11	14	Class Of Section = 4		*
99	\$WBCL1	11.A	16	Web Class = 1		*
100	\$WBCL2	11.A	16	Web Class = 2		*
101	\$WBCL3	11.A	16	Web Class = 3		*
102	\$WBCL4	11.A	16	Web Class = 4		*
110	\$FLCL1	11.B	18	Flange Class = 1		*
111	\$FLCL2	11.B	18	Flange Class = 2		*
112	\$FLCL3	11.B	18	Flange Class = 3		*



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
113	\$FLCL4	11.8	18	Flange Class = 4		*
119	\$8F			Zero Bearing Force	$B_f=0$	*
120	\$PF	7.2.2		Factored Axial Force	$P_f$	*
121	\$PFO			Zero Axial Force	$P_f=0$	*
122	\$MFX0			Zero Moment About Major A-is	$M_{fx}=0$	*
123	\$MFY0			Zero Moment About Minor Axis	$M_{fy}=0$	*
124	\$VO			Zero Shear Force	$V_f=0$	*
125	\$PFLTO		20	Axial Compression	$P_f>0$	*
126	\$PFSTO		20	Axial Tension	$P_f<0$	*
127	\$MXGTO			Moment Applied About The Minor Axis	$M_{fy}>0$	*
128	\$MXGTO			Moment Applied About The Major Axis	$M_{fx}>0$	*
129	\$BMGTO			$M_{fx}$ and $M_{fy}>0$		*
130	\$MGTO			$M_{fx}$ and/or $M_{fy}>0$		*
131	\$PRT	13.2.A.1		Factored Axial Tensile Resistance		*
132	\$MFX	7.2.2		Factored Moment About Major Axis	$P-t$	*
133	\$MFY	7.2.2		Factored Moment About Minor Axis	$M_{fx}$	*
134	\$MRX	13.6.A.1		Factored Moment Of Resistance About Major Axis	$M_{fy}$	*
135	\$MRY			Factored Moment Of Resistance About Minor Axis	$M_{rx}$	*
136	\$PC015			$P_f/C_y > 0.15$	$M_{ry}$	*
137	\$MU	13.6.2.A		Moment of Resistance Of a Member Subjected To Lateral Buckling	$M_u$	*
138	\$MP			Plastic Moment	$M_p$	*
139	\$MU23			$M_u > 2/3 M_p$		*
140	\$MF1			Smaller Moment At The Ends Of The Unbraced Length About The Major Axis	$M_{f1}$	*
141	\$MF2			Larger Moment At The Ends Of The Unbraced Length About The Major Axis	$M_{f2}$	*
142	\$MY			Yield Moment	$M_y$	*
143	\$MF1Y			Smaller Moment At The Ends Of The Unbraced Length About The Minor Axis	$M_{f1y}$	*
144	\$MF2Y			Larger Moment At The Ends Of The Unbraced Length About The Minor Axis	$M_{f2y}$	*
145	\$FRC	13.3.1.A.1		Factored Axial Compressive Resistance	$P_{rc}$	*
146	\$MU23y			$M_u > 2/3 M_y$		*
150	R1	13.2		$P_f/P_r$ Ratio		*
151	R2	X.5		$M_{fx}/M_{rx}$ Ratio		*



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
152	R3	X.5		$M_{fy}/M_{py}$ Ratio		*
153	R4			$V_f/V_r$ Ratio		*
154	\$R123			$R_1 + R_2 + R_3 \leq 1.0$		*
155	\$R4			$V_f/V_r \leq 1.0$		*
165	\$SANG		22	Single Angle		*
166	\$DANG		22	Double Angle		*
167	\$CHOLW		22	Circular Hollow Section		*
168	\$T		22	T Section		*
169	\$I		22	I Section		*
170	\$CHAL		22	Channel Section		*
171	\$REHOL		22	Rectangular Hollow Section		*
172	\$BOX		22	Box Section		*
173	\$FCOV		22	Flange Cover Plate/Diaphragm Plate		*
174	\$PCOV		22	Perforated Cover Plate		*
175	\$ISTIF		22	Plate Girder Transverse Intermediate Stiffener/Bearing Stiffener		*
176	\$CHPT			Channel Prevented From Twisting		*
177	\$DOSY			Doubly Symmetric Section		*
178	\$ASYLO			Sections Axis Of Sym. In Plane Of Loading		*
179	\$CONT			Is The Double Angle Continuously Connected By Adequate Mechanical Fasteners Or Welds?		*
180	\$DHOL			Doubly Symmetric Hollow Section (Square Hollow Section Or Circular Hollow Section)		*
185	UKX	9.3		Effective Length Factor About The Major Axis	$K_x$	*
186	ELX			Length Of Compression Or Tension Member Along The Major Axis	$L_x$	*
187	UFL			Unsupported Length Of Compression Flange	$L$	*
188	SRY			Radius Of Gyration Of Member About Its Minor Axis	$r_y$	*
189	SRT			Radius Of Gyration About Its Axis Of Symmetry Of A Tee Section Comprising The Compression Flange and 1/6 Of Web	$r_t$	*
190	CY			Axial Compressive Load At Yield Stress	$C_y$	*
191	SKB			Plate Buckling Coefficient	$K_b$	*
192	FIY			Moment Of Inertia About The Minor Axis	$I_y$	*
193	FIX			Moment Of Inertia About The Major Axis	$I_x$	*





DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
194	OMEGAX	13.6.2.A.1		Coefficient Used To Determine Equivalent Uniform Bending Effect In Beam Columns Or Coefficient In Column Equation About The Major Axis	$\omega_x$	*
195	SIGMA1	13.6.1.A.1		$\sigma_1 = 20000/(Ld/A_f)$	$\sigma_1$	*
196	SIGMA2	13.6.1.A.1		$\sigma_2 = 250000/(L/r_c)^2$	$\sigma_2$	*
197	FAI1			Performance Factor	$\phi$	*
198	S(1)			Pitch Between Successive Fastener Holes Along Critical Section		*
199	S(2)			Pitch Between Successive Fastener Holes Along Critical Section		*
200	S(3)			Pitch Between Successive Fastener Holes Along Critical Section		*
201	S(4)			Pitch Between Successive Fastener Holes Along Critical Section		*
202	G(1)			Gauge Distance Between Fasteners Along Critical Section		*
203	G(2)			Gauge Distance Between Fasteners Along Critical Section		*
204	G(3)			Gauge Distance Between Fasteners Along Critical Section		*
205	G(4)			Gauge Distance Between Fasteners Along Critical Section		*
208	SRX			Radius Of Gyration Of Member About Its Major Axis	$r_x$	*
211	TSLRAX	13.2		Tension Member Slenderness Ratio About The Major Axis		*
212	TSLRAY	13.2		Tension Member Slenderness Ratio About The Minor Axis		*
213	TLR	13.2		Maximum of (TSLRAX, TSLRAY)		*
214	UKY	9.3		Effective Length Factor About The Minor Axis	$K_y$	*
215	UKLR	9.3		Maximum of (SLRAX, SLRAY) In Compression Member		*
216	ELY			Length Of Compression Or Tension Member Along The Minor Axis	$L_y$	*
217	SLRAY	9.3		Compression Member Slenderness Ratio About The Minor Axis		*
218	REDUTM	12.3.A.1		$\Sigma S_K^2/4g_K$		*
219	SLRAX	9.3		Compression Member Slenderness Ratio About The Major Axis		*
220	\$SWYX			Sway Effects About The Major Axis Included In Analysis/ Sway Effects About The Major Axis Resisted By Bracing Or Shear Wall		*
221	\$SWYY			Sway Effects About The Minor Axis Included In Analysis/ Sway Effects About The Minor Axis Resisted By Bracing Or Shear Wall		*
222	\$ROTX			Rotational Restraint At The End Of The Unbraced Lengths Shows K Can Be < 1.0 Along The Major Axis		*
223	\$ROTY			Rotational Restraint At The End Of The Unbraced Lengths Shows K Can Be < 1.0 Along The Minor Axis		*
224	\$WBACM		26	Web In Axial Compression		
225	\$WBFLX		26	Web In Flexural Compression		



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
226	\$WBC08		26	Web-In Combine Axial And Flexural Compression		*
230	\$NMGA	13.3.1		Non-Dimensional Slenderness Ratio In Column Formula	$\lambda$	*
231	\$NMGA1		28	$0 \leq \lambda \leq 1.0$		*
232	\$NMGA2		28	$1.0 < \lambda \leq 2.0$		*
233	\$NMGA3		28	$2.0 < \lambda \leq 3.6$		*
234	\$NMGA4		28	$3.6 < \lambda$		*
236	\$DT1		30	$D1/t \leq 260/\sqrt{F_y}$		*
237	\$DT2		30	$260/\sqrt{F_y} < D1/t \leq 365/\sqrt{F_y}$		*
238	\$DT3		30	$365/\sqrt{F_y} < D1/t \leq 3300/F_y$		*
239	\$DT4		30	$D1/t > 3300/F_y$		*
241	\$BT2		32	$b/t \leq 75/\sqrt{F_y}$		*
242	\$BT2		32	$b/t > 75/\sqrt{F_y}$		*
243	\$BT3		34	$b/t \leq 100/\sqrt{F_y}$		*
244	\$BT4		34	$b/t > 100/\sqrt{F_y}$		*
245	\$BT5		36	$b/t \leq 200/\sqrt{F_y}$		*
246	\$BT6		36	$200/\sqrt{F_y} < b/t \leq 255/\sqrt{F_y}$		*
247	\$BT7		36	$b/t > 255/\sqrt{F_y}$		*
248	\$BT8		38	$b/t \leq 320/\sqrt{F_y}$		*
249	\$BT9		38	$b/t > 320/\sqrt{F_y}$		*
250	\$BT10		40	$b/t \leq 160/\sqrt{F_y}$		*
251	\$BT11		40	$160/\sqrt{F_y} < b/t \leq 200/\sqrt{F_y}$		*
252	\$BT12		40	$200/\sqrt{F_y} < b/t \leq 255/\sqrt{F_y}$		*
253	\$BT13		40	$b/t > 255/\sqrt{F_y}$		*
254	\$BT14		42	$b/t \leq 54/\sqrt{F_y}$		*
255	\$BT15		42	$54/\sqrt{F_y} < b/t \leq 64/\sqrt{F_y}$		*



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
256	\$BT16		42	$64/\sqrt{F_y} < b/t \leq 100/\sqrt{F_y}$		*
257	\$BT17		42	$b/t > 100/\sqrt{F_y}$		*
258	\$BT18			$b/t \leq 201/\sqrt{R_b/F_y}$		*
260	\$hW1		44	$h/w \leq 255/\sqrt{F_y}$		*
261	\$hW2		44	$h/w < 255/\sqrt{F_y}$		*
262	\$hW3		46	$h/w \leq 420/\sqrt{F_y}$		*
263	\$hW4		46	$420/\sqrt{F_y} < h/w \leq 520/\sqrt{F_y}$		*
264	\$hW5		46	$520/\sqrt{F_y} < h/w \leq 690/\sqrt{F_y}$		*
265	\$hW6		46	$h/w > 690/\sqrt{F_y}$		*
266	\$hW7		48	$h/w \leq 420/\sqrt{F_y} (1-1.4 \frac{P_f}{C_y})$		*
267	\$hW8		48	$420/\sqrt{F_y} (1-1.4 \frac{P_f}{C_y}) < h/w \leq 450/\sqrt{F_y} (1-0.43 \frac{P_f}{C_y})$		*
268	\$hW9		48	$h/w > 450/\sqrt{F_y} (1-0.43 \frac{P_f}{C_y})$		*
269	\$hW10		50	$420/\sqrt{F_y} (1-1.4 \frac{P_f}{C_y}) < h/w \leq 520/\sqrt{F_y} (1-1.28 \frac{P_f}{C_y})$		*
270	\$hW11		50	$520/\sqrt{F_y} (1-1.28 \frac{P_f}{C_y}) < h/w \leq 690/\sqrt{F_y} (1-2.6 \frac{P_f}{C_y})$		*
271	\$hW12		50	$h/w > 690/\sqrt{F_y} (1-2.6 \frac{P_f}{C_y})$		*
272	\$hW13		52	$h/w \leq 690/\sqrt{F_y}$		*
273	\$hW14		52	$690/\sqrt{F_y} < h/w \leq 690/\sqrt{45F_{cr}/\phi_s}$		*
274	\$hW15		52	$690/\sqrt{45F_{cr}/\phi_s} < h/w \leq 12000/F_y$		*
275	\$hW16			$h/w > 12000/F_y$		*
276	\$hW17		50	$h/w \leq 420/\sqrt{F_y} (1-1.4 \frac{P_f}{C_y})$		*
280	\$FYFU1		54	$F_y/F_u \leq 0.75$		*



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
281	\$FYU2		54	$0.75 < F_y/F_u \leq 0.85$		*
282	\$FYU3		54	$0.85 < F_y/F_u$		*
283	\$ANG1			$A_n/A_g \geq F_y/F_u$		*
284	\$HW17			$h/w \leq 167\sqrt{K_v/F_y}$		*
285	\$HW18			$167\sqrt{K_v/F_y} < h/w \leq 190\sqrt{K_v/F_y}$		*
286	\$HW19			$190\sqrt{K_v/F_y} < h/w \leq 239\sqrt{K_v/F_y}$		*
287	\$HW20			$239\sqrt{K_v/F_y} < h/w$		*
290	\$CNB		56	Axial Compression And Bending		*
291	\$TNB		56	Axial Tension And Bending		*
295	FMRX1	13.8.A.1		Factored Moment Of Resistance About The Major Axis Calculated By Clause 13.5		*
296	FMRV1	13.8.A.1		Factored Moment Of Resistance About The Minor Axis Calculated By Clause 13.5	M <sub>rx1</sub>	*
297	FPRC1	13.8.A.6		Factored Axial Compressive Resistance Used In Data 337 and Data 339	M <sub>ry1</sub>	*
298	FPRC2	13.8.A.8		Factored Axial Compressive Resistance Used In Data 336 and Data 338	P <sub>rc1</sub>	*
299	FMRX2	13.8.A.3		Factored Moment Of Resistance About Major Axis Defined In Clause 13.6	P <sub>rc2</sub>	*
300	FMR1	13.9.A.1		Factored Moment Of Resistance About Both Axis In Axial Tension And Bending	M <sub>rx2</sub>	*
310	OMEGAY	13.8.A.10		Coefficient Used To Determine Equivalent Uniform Bending Effect In Beam Columns Or Coefficient In Column Equation About The Minor Axis	M <sub>r1</sub>	*
311	CEX	13.8.A.9		Euler Buckling Load About The Major Axis	$\omega_y$	*
312	CEY	13.8.A.9		Euler Buckling Load About The Minor Axis	C <sub>ex</sub>	*
313	\$NMGA1	13.8.A.7		Non Dimensional Slenderness Ratio In Column Formula For Beam Column	C <sub>ey</sub>	*
314	\$NMGA5		58	$0 \leq \lambda_1 \leq 1.0$	$\lambda_1$	*
315	\$NMGA6		58	$1.0 < \lambda_1 \leq 2.0$		*
316	\$NMGA7		58	$2.0 < \lambda_1 \leq 3.6$		*
317	\$NMGA8		58	$3.6 < \lambda_1$		*
320	\$LOAD4		60	Member Not Subjected To Transverse Loads In The Major Axis Plane Between Supports		*
321	\$LOAD5		60	Member Subjected To Distributed Load Or Series Of Point Loads In The Major Axis Plane Between Supports		*





DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
322	\$LOAD6		60	Member Subjected To A Concentrated Load Or Moment In The Major Axis Plane Between Supports		*
323	\$CURV3		62	Member Bent In Single Curvature About The Minor Axis		*
324	\$CURV4		62	Member Bent In Double Curvature About The Minor Axis		*
325	VF	7.2.2		Shear Force In A Member Or Component Under Factored Load	Vf	*
326	VR	13.4.A.1		Factored Shear Resistance In A Member Or Component	VR	*
327	FS	13.4.1		Ultimate Shear Strength	FS	*
328	SKV	13.4.1.A.1		Shear Buckling Coefficient	Kv	*
329	T	13.4.1.A.1		Non-Dimensional Coefficient In F <sub>s</sub> Formula	T	*
330	SNAGA	13.4.1.A.1		Non-Dimensional Coefficient In F <sub>s</sub> Formula	η	*
331	SA			Distance Between Stiffeners	a	*
332	\$AH1		64	a/h < 1.0		*
333	\$AH2		64	a/h ≥ 1.0		*
334	\$STW			Stiffened Webs		*
335	\$R5			$\frac{M_{fx}}{M_{rx1}} + \frac{M_{fy}}{M_{ry1}} \leq 1.0$		*
336	\$R6			$\frac{P_f}{P_{rc2}} + \frac{0.85 M_{fx}}{M_{rx1}} + \frac{0.6 M_{fy}}{M_{ry1}} \leq 1.0$		*
337	\$R7			$\frac{P_f}{P_{rc1}} + \frac{\omega_x M_{fx}}{M_{rx2}(1-\frac{P_f}{C_{ey}})} + \frac{\omega_y M_{fy}}{M_{ry1}(1-\frac{P_f}{C_{ey}})} \leq 1.0$		*
338	\$R8			$\frac{P_f}{P_{rc2}} + \frac{M_{fx}}{M_{rx1}} + \frac{M_{fy}}{M_{ry1}} \leq 1.0$		*
339	\$R9			$\frac{P_f}{P_{rc1}} + \frac{\omega_x M_{fx}}{M_{rx2}(1-\frac{P_f}{C_{ex}})} + \frac{\omega_y M_{fy}}{M_{ry1}(1-\frac{P_f}{C_{ey}})} \leq 1.0$		*



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
340	\$R10			$\frac{P_f}{P_{rt}} + \frac{M_{fx}}{M_{rt}} + \frac{M_{fy}}{M_{rt}} \leq 1.0$		
341	\$R11			$\frac{M_{fx}}{M_{rx2}} + \frac{M_{fy}}{M_{ry1}} - \frac{P_f}{P_{rt}} \leq 1.0$		
342	\$FLSH		66	Flexural Member Subjected To Shear		*
343	\$GUSP		66	Gusset Plates		*
344	\$PINS		66	Pins		*
350	\$PLA1			Web Stiffener Are Supplied On The Member At A Point Of Load Application Where A Plastic Hinge Would Form		
351	\$PLA2			Splices In The Member Are Designed To Transmit 1.1 Times The Max. Computed Moment Under Factor Loads At The Splice Location Or 0.25 $M_p$ Whichever Is Greater		
352	\$PLA3			Member Is Not Subjected To Repeated Impact Or Fatigue		
353	\$PLA4			The Influence Of Inelastic Deformation On The Strength Of The Structure Shall Be Taken Into Account		
354	\$F08			$F_y \leq 0.8 F_u$		
355	ELP			Laterally Unbraced Length Between Plastic Hinges	$L_{ph}$	*
356	ELCR	13.7		Max. Unbraced Length Adjacent To A Plastic Hinge	$L_{cr}$	*
357	\$L1			$ELP \leq ELCR$		
358	\$MF1		68	$M_{f1}/M_{f2} > 0.5$		*
359	\$MF2		68	$M_{f1}/M_{f2} \leq 0.5$		*
360	\$COV			Cover Plate Used		*
361	\$BGIR		70	Bolted Girder		
362	\$WGIR		70	Welded Girder		
363	\$ACOV			Area Of Cover Plate	$A_{cov}$	*
364	\$PCOV			Force In Cover Plate	$P_{cov}$	*
365	\$MFC			Moment Due To Factored Load At Point Of Theoretical Cut Off	$M_{fc}$	*
366	Y			Distance From Centroid Of Cover Plate To Neutral Axis Of Cover Plated Section	$y$	
367	\$FICOV			Moment of Inertia Of Cover Plated Section	$I_{cov}$	*
368	\$A			$A_{cov} \leq 70\% A_f$		



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
369	\$PCOV			$P_{cov} \geq (A_{cov} \cdot M_{fc} \cdot V) / I_{cov}$		*
370	\$IBST		72	Interior Bearing Stiffener		*
371	\$EBST	15.5	72	End Bearing Stiffener		*
372	\$AWEb			Equivalent Web Area To B - Included In Area Of Stiffener	A <sub>web</sub>	*
373	\$ABST			Area Of Bearing Stiffener	A <sub>bst</sub>	*
374	\$FIYWB			Moment Of Inertia Of Intermediate Stiffener Or Pair Of Stiffeners About An Axis In The Plane Of The Web	I <sub>ywb</sub>	*
375	\$AIST			Area Of Intermediate Stiffener	A <sub>ist</sub>	*
376	C	15.6.A.1		$C = 1 - (45000 K_y) / F_y (h/w)^2$	C	*
377	YRAT	15.6.A.1		Ratio Of Specified Min. Yield Strength Of Web Steel To Specified Min. Yield Strength Of Flange Steel	YRAT	*
378	SFACT	15.6.A.2		Stiffener Factor	SFACT	*
379	REFAC	15.6.A.3		Reduction Factor $V_f/V_r$		*
380	VC			Shear Capacity Of Weld Connection In (k/in) Between The Web And The Stiffener	V <sub>c</sub>	*
381	TLOAD			Total Load Required To Be Transmitted To The Web Through The Stiffener	TLOAD	*
382	\$IST1			$I_{ywb} \geq ((h/50)^4)$		*
383	\$IST2			$A_{1st} \geq aw/2 [1 - \frac{a/h}{\sqrt{1+(a/h)^2}}] C \times YRAT \times SFACT \times REFAC$		*
384	\$IST3			$V_c \geq \max (0.0026 h F_y^{3/2} \times REFAC, TLOAD/h)$		*
385	FYSTIF			F <sub>y</sub> Of Stiffener Steel	F <sub>y stiff</sub>	*
386	\$IST4		74	Intermediate Stiffener Furnished In Pairs		*
387	\$IST5		74	Intermediate Single Angle Stiffener		*
388	\$IST6		74	Intermediate Single Plate Stiffener		*
389	VFADJ			Largest Factored Shear In An Adjacent Panel	V <sub>fadj</sub>	*
390	\$IST7			$V_{fadj}/V_r < 1.0$		*
391	BR1	13.9.A.1		Bearing Resistance Of Stiffener	B <sub>r1</sub>	*
392	SLOAD			Sum Of Loads Not Supported By A Bearing Stiffener On The Compressive Edge Of Web Plate	SLOAD	*
393	\$BL1			$B_{r1} \geq SLOAD$		*
394	\$FLR0		76	Flange Not Restrained Against Rotation		*



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
395	\$FLNR0		76	Flange Restrained Against Rotation		*
396	ATW			A In Clause 15.9	ATW	*
397	\$BST1			$K_{Lx} > 3/4 L_x$		*
398	\$BST2			$K_{Ly} > 3/4 L_y$		*
400	GAMA	7.2.5		Importance Factor	Y	*
401	ALFAD	7.2.3		Load Factor For Dead Load	$\phi_D$	*
402	ALFAL	7.2.3		Load Factor For Live Load	$\phi_L$	*
403	ALFAQ	7.2.3		Load Factor For Earthquake Load	$\phi_Q$	*
404	ALFAT	7.2.3		Load Factor For Temperature Effect	$\phi_T$	*
405	SIYE	7.2.4		Load Combination Factor	$\psi$	*
406	FPD			Axial Load Due To Dead Load	PD	*
407	FPL			Axial Load Due To Live Load	PL	*
408	FPQ			Axial Load Due To Earthquake	PQ	*
409	FPT			Axial Load Due To Temperature Effect	PT	*
410	FMXD			Moment About Major Axis Due To Dead Load	MXD	*
411	FMXL			Moment About Major Axis Due To Live Load	MXL	*
412	FMXQ			Moment About Major Axis Due To Earthquake Load	MXQ	*
413	FMXT			Moment About Major Axis Due To Temperature Effect	MXT	*
414	FMXD			Moment About Minor Axis Due To Dead Load	MYD	*
415	FMYL			Moment About Minor Axis Due To Live Load	MYL	*
416	FMYQ			Moment About Minor Axis Due To Earthquake Load	MYQ	*
417	FMYT			Moment About Minor Axis Due To Temperature Effect	MYT	*
418	VD			Shear Force Due To Dead Load	VD	*
419	VL			Shear Force Due To Live Load	VL	*
420	VQ			Shear Force Due To Earthquake Load	VQ	*
421	VT			Shear Force Due To Temperature Effect	VT	*
422	\$FACT1		78	Overturning, Uplift, Or Stress Reversal Case		*
423	\$FACT2			Only One Of L, T, Q Act		*
424	\$FACT3		78	Two Of L, T, Q Act		*
425	\$FACT4		78	All Of L, T, Q Act		*
426	\$FACT5			Farm Building Of Low Occupancy Rate/ /Building Collapse Not Likely To Cause Injury		*





DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
427	BJ			Bearing Load Due To Dead Load	B <sub>D</sub>	*
428	BL			Bearing Load Due To Live Load	B <sub>L</sub>	*
429	BQ			Bearing Load Due To Earthquake Load	B <sub>Q</sub>	*
430	BT			Bearing Load Due To Temperature Effect	B <sub>T</sub>	*
431	BF	7.2.2		Factored Bearing Load	B <sub>F</sub>	*
432	BR	13.10.A.1		Factored Bearing Resistance Of A Member Or Component	B <sub>R</sub>	*
433	ACONT			Area Of Contact In Bearing	A <sub>contact</sub>	*
434	RD			Diameter Of Roller Or Rocker	D	*
435	RL			Length Of Roller Or Rocker	L	*
436	\$BRA		80	On The Contact Area Of Machined, Accurately Sawn Or Thread Parts		*
437	\$BRB		80	On Expansion Rollers Or Rockers		*
438	RB	13.10.A.1		$B_F/B_R$		*
439	\$RB			$RB \leq 1.0$	R <sub>B</sub>	
440	SY			Section Modulus About the Weak Axis	s <sub>y</sub>	
441	ZY			Plastic Modulus About the Weak Axis	z <sub>y</sub>	



DATA NUMBER	DATA NAME	TABLE ADDRESS	SET NO.	DATA DESCRIPTION	DATA SYMBOL	DATA DEPD.
500	CHECKN	13.8.R		The Number Of Times A Table Is To Be Recycled		
501	CHECKI	X.6.R		Counter For Recycling Of Table		
502	FMRXA	X.7.R		Factored Moment Of Resistance About The Major Axis	$M_{rYA}$	
503	FMRYA	X.7.R		Factored Moment Of Resistance About The Minor Axis	$M_{rYA}$	
504	FPRCA	X.7.R		Factored Compressive Resistance	$P_{rCA}$	
505	OMGXA	X.7.R			$\omega_{xA}$	
506	OMGYA	X.7.R			$\omega_{yA}$	
507	THETAX	X.7.R			$\theta_x$	
508	THETAY	X.7.R			$\theta_y$	
509	\$RATIO			$\frac{1-P_f/C_{ex}}{1-P_f/C_{ey}} + \frac{M_{rx} \omega_{xA}}{P_{rCA} M_{rYA} \theta_x} + \frac{M_{ry} \omega_{yA}}{M_{rYA} \theta_y} \leq 1.0$		
510	\$CK1		82	CHECKI=1		
511	\$CK2		82	CHECKI=2		
512	\$CK3		82	CHECKI=3		
513	\$CN1		84	CHECKN=2		
514	\$CN2		84	CHECKN=3		
515	\$RAT11			$\frac{M_{fx}}{M_{rXA}} + \frac{M_{fy}}{M_{rYA}} \leq 1.0$		
516	\$RAT12			$\frac{P_f}{P_{rCA}} + \frac{M_{rx} \omega_{xA}}{M_{rXA} \theta_x} + \frac{M_{ry} \omega_{yA}}{M_{rYA} \theta_y} \leq 1.0$		







# NON COMPACTED LIST OF DEPENDENTS

DATA NO. OF	LIST OF DEPENDENTS	
NO. DEPENDENTS		
83 (1)	137 134 151 152 135	295 296 300 299
95 (9)	230 134 151 152 135	295 296 300 299
96 (9)	230 134 151 152 135	295 296 300 299
97 (9)	230 134 151 152 135	295 296 300 299
98 (9)	230 134 151 152 135	295 296 300 299
99 (4)	97 98 134 151	
100 (4)	97 98 134 151	
101 (4)	97 98 134 151	
102 (4)	97 98 134 151	
110 (6)	97 98 134 151 295	296
111 (6)	97 98 134 151 295	296
112 (6)	97 98 134 151 295	296
113 (6)	97 98 134 151 295	296
114 (6)	97 98 134 151 295	296
120 (15)	156 151 153 152 133	132 325 120 431
121 (9)	136 266 267 268 276	269 270 271 150 336
122 (9)	150 151 153 152 133	132 325 120 431
123 (9)	150 151 153 152 133	132 325 120 431
124 (9)	150 151 153 152 133	132 325 120 431
125 (7)	95 96 97 211 212	213 150
126 (7)	95 96 97 211 212	213 150
127 (1)	135	
128 (1)	135	
129 (1)	135	
130 (3)	95 96 97	
131 (3)	150 340 341	
132 (6)	151 335 336 337 338	339 340 341
133 (7)	152 335 336 338 339	340 341
134 (2)	151 135	
135 (1)	152	
136 (4)	139 134 146 299	
137 (4)	139 134 146 299	300
138 (6)	139 134 146 299 299	
139 (3)	134 151 299	
140 (3)	194 358 359	
141 (3)	194 358 359	
142 (4)	146 134 299 300	
143 (1)	310	
144 (1)	310	
145 (1)	150	
146 (3)	134 151 299	
150 (1)	154	
151 (1)	154	
152 (1)	154	
153 (9)	110 111 112 113 97	98 137 295 296
154 (9)	110 111 112 113 97	98 137 295 296
155 (9)	110 111 112 113 97	98 137 295 296
156 (9)	110 111 112 113 97	98 137 295 296
157 (9)	110 111 112 113 97	98 137 295 296
158 (9)	110 111 112 113 97	98 137 295 296
159 (9)	110 111 112 113 97	98 137 295 296
160 (9)	110 111 112 113 97	98 137 295 296
161 (9)	110 111 112 113 97	98 137 295 296
162 (9)	110 111 112 113 97	98 137 295 296
163 (9)	110 111 112 113 97	98 137 295 296
164 (9)	110 111 112 113 97	98 137 295 296
165 (9)	110 111 112 113 97	98 137 295 296
166 (9)	110 111 112 113 97	98 137 295 296
167 (9)	110 111 112 113 97	98 137 295 296
168 (9)	110 111 112 113 97	98 137 295 296
169 (9)	110 111 112 113 97	98 137 295 296
170 (9)	110 111 112 113 97	98 137 295 296
171 (9)	110 111 112 113 97	98 137 295 296





## NON COMPACTED LIST OF DEPENDENTS

DATA NO.	NO. OF DEPENDENTS	LIST OF DEPENDENTS	DATA NO.	NO. OF DEPENDENTS	LIST OF DEPENDENTS
172	( 9 )	110 111 112 113 97	242	( 2 )	97 98
173	( 9 )	110 111 112 113 97	243	( 4 )	97 98 112 113
174	( 9 )	110 111 112 113 97	244	( 4 )	97 98 112 113
175	( 9 )	110 111 112 113 97	245	( 6 )	95 97 98 110 112
176	( 2 )	95 96 97	246	( 6 )	95 97 98 110 112
177	( 3 )	95 96 97	247	( 6 )	95 97 98 110 112
178	( 3 )	97	248	( 2 )	97 98
179	( 1 )	135 295 296	249	( 2 )	97 98
180	( 3 )	219 230 311 397	250	( 4 )	110 111 112 113
185	( 4 )	219 211 230 311 397	251	( 4 )	110 111 112 113
186	( 5 )	137 195 196	252	( 4 )	110 111 112 113
187	( 3 )	217 212 230 312 356	253	( 4 )	110 111 112 113
188	( 5 )	196	254	( 6 )	110 111 112 113
189	( 1 )	136 266 267 268 276	255	( 6 )	110 111 112 113
190	( 8 )	258 75	256	( 6 )	110 111 112 113
191	( 2 )	137	257	( 6 )	110 111 112 113
192	( 1 )	137 337 339	258	( 1 )	75
194	( 3 )	137	259	( 2 )	99 102
195	( 1 )	137	260	( 2 )	99 102
196	( 1 )	137	261	( 2 )	99 100 101 102
197	( 3 )	131 145 134 135 295	262	( 4 )	99 100 101 102
198	( 1 )	218	263	( 4 )	99 100 101 102
199	( 1 )	218	264	( 4 )	99 100 101 102
200	( 1 )	218	265	( 4 )	99 100 101 102
201	( 1 )	218	266	( 3 )	99 100 102
203	( 1 )	218	267	( 3 )	99 100 102
204	( 1 )	218	268	( 3 )	99 100 102
205	( 1 )	218	269	( 4 )	99 100 101 102
208	( 3 )	219 211 311	270	( 4 )	99 100 101 102
211	( 1 )	213	271	( 4 )	99 100 101 102
212	( 1 )	213	272	( 4 )	134 151 295 296
214	( 2 )	217 312	273	( 4 )	134 151 295 296
215	( 1 )	217 312	274	( 4 )	134 151 295 296
216	( 3 )	217 212 312	276	( 4 )	99 100 101 102
217	( 1 )	215	280	( 1 )	45
218	( 1 )	45	281	( 1 )	45
219	( 1 )	215	282	( 1 )	45
220	( 6 )	219 217 215 214 185	283	( 2 )	131 150
221	( 6 )	219 217 215 214 185	284	( 1 )	327
222	( 5 )	219 217 215 214 185	285	( 1 )	327
223	( 5 )	219 217 215 214 185	286	( 1 )	327
224	( 5 )	231 232 233 234 185	287	( 1 )	327
230	( 5 )	145 150	288	( 3 )	335 336 338
231	( 2 )	145 150	296	( 6 )	335 336 337 338 339
232	( 2 )	145 150	297	( 2 )	337 339
233	( 2 )	145 150	298	( 2 )	337 339
234	( 2 )	145 150	299	( 5 )	340 338
236	( 4 )	95 96 97 98	300	( 1 )	340
237	( 4 )	95 96 97 98	301	( 2 )	337 339
238	( 4 )	95 96 97 98	311	( 2 )	337 339
239	( 4 )	95 96 97 98	312	( 2 )	337 339
241	( 2 )	97 98			
			296 299 297 298 300	326 391 432	
			98 137 295 296		
			98 137 295 296		
			98 137 295 296		
			269 270 271		
			194		
			310		
			341		



## NON COMPACTED LIST OF DEPENDENTS

DATA NO.	NO. OF DEPENDENTS	LIST OF DEPENDENTS
313	(5)	314 315 316 317 297
314	(1)	297
315	(1)	297
316	(1)	297
317	(1)	297
320	(1)	310
321	(1)	310
322	(1)	310
323	(1)	310
324	(1)	310
325	(1)	155
326	(3)	155 390 379
327	(1)	326
328	(1)	284
329	(1)	284 285 286 287 327 376
330	(1)	327
331	(7)	332 333 328 329 330 383 391
332	(3)	328 329 330
333	(3)	328 329 330
334	(3)	328 329 330
342	(1)	326
343	(1)	326
344	(1)	326
348	(1)	326
355	(1)	357
356	(1)	357
358	(1)	356
359	(1)	356
363	(2)	368 369
364	(1)	369
365	(1)	369
366	(1)	369
367	(1)	369
370	(2)	372 46
371	(2)	372 46
372	(1)	46
373	(1)	46
374	(1)	382
375	(1)	383
376	(1)	383
377	(1)	383
378	(1)	383
379	(2)	383 384
380	(1)	384
381	(1)	377
385	(1)	377
386	(1)	378
387	(1)	378
388	(1)	378
389	(2)	390 379
390	(1)	379

## NON COMPACTED LIST OF DEPENDENTS

DATA NO.	NO. OF DEPENDENTS	LIST OF DEPENDENTS
391	(1)	393
392	(1)	393
394	(1)	391
395	(1)	391
396	(1)	391
400	(5)	120 132 133 325 431
401	(5)	120 132 133 325 431
402	(5)	120 132 133 325 431
403	(5)	120 132 133 325 431
404	(5)	120 132 133 325 431
405	(5)	120 132 133 325 431
406	(1)	120
407	(1)	120
408	(1)	120
409	(1)	120
410	(1)	132
411	(1)	132
412	(1)	132
413	(1)	132
414	(1)	133
415	(1)	133
416	(1)	133
417	(1)	133
418	(1)	325
419	(1)	325
420	(1)	325
421	(1)	325
422	(4)	401 402 403 404
423	(1)	405
424	(1)	405
425	(1)	405
426	(1)	400
427	(1)	431
428	(1)	431
429	(1)	431
430	(1)	431
431	(1)	438
432	(1)	438
433	(1)	432
434	(1)	432
435	(1)	432
436	(2)	432 438
437	(2)	432 438



DECISION TABLE X.1 (1)

Data Requirement

Elastic Analysis Plastic Analysis	x x	
Elastic Analysis Plastic Analysis		Y N N Y
Execute Table X.2 Execute Table 8.5		Y Y

DECISION TABLE X.2 (2)

Data Requirement

Structural Steel Member Open Web Steel Joist Composite Construction Member Built Up Member Connection Girder Bearing Stiffener Intermediate Transverse Stiffener	x x x x x x x x	
Structural Steel Member Open-Web Steel Joists Composite Construction Member Built-Up Members Connections Girder Bearing Stiffener Intermediate Transverse Stiffener	Y N N Y N N N Y N N N N N N N N	N N N N N N Y N N N N Y N Y N Y
Execute Table X.3 Execute Table — Execute Table — Execute Table — Execute Table 15. Execute Table 15.5 Execute Table 15.6	Y Y Y Y Y Y Y Y	N N N Y N N N N









DECISION TABLE 9.3 (4)

Data Requirement

Compression Member		
Compression Member in Truss	X	
Sway Effects About The Major Axis Included in Analysis/ / Sway Effects About the Major Axis Resisted by Bracing or Shear Wall	X	
Sway Effects About the Minor Axis Included in Analysis/ / Sway Effects About the Minor Axis Resisted by Bracing or Shear Wall	X	
Rotational restraints at the ends of the unbraced length show $k$ can be $< 1.0$ About the Major Axis	X	Table 9.3 (4)
Rotational restraints at the ends of the unbraced length show $k$ can be $< 1.0$ about the Minor Axis	X	Table 9.3 (4)
$K_x$		
$K_y$		
$L_x$	X	
$r_x$	X	
$r_y$	X	
SLRAX		Table 9.3 (4)
SLRAY		Table 9.3 (4)







TABLE 11. (5)

Data Requirement

Web Class = 1	Table 11.A (6)
Web Class = 2	Table 11.A (6)
Web Class = 3	Table 11.A (6)
Web Class = 4	Table 11.A (6)
Flange Class = 1	Table 11.B (20)
Flange Class = 2	Table 11.B (20)
Flange Class = 3	Table 11.B (20)
Flange Class = 4	Table 11.B (20)
Circular Hollow Section	
Single Angle	x
Double Angle	x
Flange Cover Plate/Diaphragm Plate	x
Perforated Cover Plate	x
Plate Girder Intermediate	x
Transverse Stiffener/Bearing Stiffener	x









DECISION TABLE 11.A (6)

Data Requirement

I Section	x
I Section	x
I Section	x
Rectangular Hollow Section	x
Box Section	x

I Section	Y	N	N	N	N
I Section	Y	N	N	N	N
I Section	N	N	N	Y	N
Rectangular Hollow Section	N	N	N	Y	N
Box Section	N	N	N	N	Y
Execute Table 11.A.8	Y	Y	Y	Y	Y

DECISION TABLE 11.A.1 (7)

Data Requirement

$P_f > 0$	x
$M_{fx}$ and/or $M_{fy} > 0$	x
Section Doubly Symmetric	x
Section's Axis of Symmetry in Plane of Loading	x

$P_f > 0$	N	Y	N	N	Y	Y
$M_{fx}$ and/or $M_{fy} > 0$	N	N	N	Y	Y	Y
Section Doubly Symmetric	I	Y	N	N	N	N
Section's Axis of Sym. in Plane of Loading	I	I	I	N	I	N
Class = 1	Y	Y	Y	Y	Y	Y
Class = 2	Y	Y	Y	Y	Y	Y



DECISION TABLE 11.A.2 (8)

Data Requirement

$P_f > 0$ $M_{fx}$ and/or $M_{fy} > 0$ Section's Axis of Symmetry in Plane of Loading	x x x
--	-------------

$P_f > 0$ $M_{fx}$ and/or $M_{fy} > 0$ Section's Axis of Sym. in Plane of Loading	N Y N N Y Y N N Y Y Y Y I I Y N Y N
Class = 2 Class = 3	Y Y Y Y Y Y Y Y

DECISION TABLE 11.A.3 (9)

Data Requirement

D1 t Fy	x x x
---------------	-------------

$\frac{250/\sqrt{F_y}}{365/\sqrt{F_y}} < \frac{D1/t}{D1/t} \leq \frac{260/\sqrt{F_y}}{365/\sqrt{F_y}}$ $D1/t < 260/\sqrt{F_y}$ $D1/t \leq 365/\sqrt{F_y}$ $D1/t < 3300/F_y$ $D1/t > 3300/F_y$	Y N N N N Y N N N N Y N N N N Y	D1 t Fy D1 t Fy D1 t Fy D1 t Fy
Class = 1 Class = 2 Class = 3 Class = 4	Y Y Y Y	



DECISION TABLE 11.A.4 (10)

Data Requirement

b	x	
t	x	
F <sub>y</sub>	x	

$b/t < 75/\sqrt{F_y}$ $b/t > 75/\sqrt{F_y}$	Y N N Y	b t F <sub>y</sub> b t F <sub>y</sub>
Class = 3 Class = 4	Y Y	

DECISION TABLE 11.A.5 (11)

Data Requirement

b	x	
t	x	
F <sub>y</sub>	x	

$b/t \leq 100/\sqrt{F_y}$ $b/t > 100/\sqrt{F_y}$	Y N N Y	b t F <sub>y</sub> b t F <sub>y</sub>
Class = 3 Class = 4	Y Y	



DECISION TABLE 11.A.6 (12)

Data Requirement

b	x	
t	x	
F <sub>y</sub>	x	

$\begin{aligned} &b/t \leq 200/\sqrt{F_y} \\ &200/\sqrt{F_y} < b/t \leq 255/\sqrt{F_y} \\ &b/t > 255/\sqrt{F_y} \end{aligned}$	$\begin{aligned} &Y \ N \ N \\ &N \ Y \ N \\ &N \ N \ Y \end{aligned}$	$\begin{aligned} &b \ t \ F_y \\ &b \ t \ F_y \\ &b \ t \ F_y \end{aligned}$
Class = 1 Class = 3 Class = 4	$\begin{aligned} &Y \\ &Y \\ &Y \end{aligned}$	

DECISION TABLE 11.A.7 (13)

Data Requirement

b	x	
t	x	
F <sub>y</sub>	x	

$\begin{aligned} &b/t \leq 320/\sqrt{F_y} \\ &b/t > 320/\sqrt{F_y} \end{aligned}$	$\begin{aligned} &Y \ N \\ &N \ Y \end{aligned}$	$\begin{aligned} &b \ t \ F_y \\ &b \ t \ F_y \end{aligned}$
Class = 3 Class = 4	$\begin{aligned} &Y \\ &Y \end{aligned}$	





DECISION TABLE 11.A.8 (14)

Data Requirement

Web in Axial Compression	x
Web in Flexural Compression	x
Web in Combine Axial and Flexural Compression	x
h	x
w	x
F <sub>y</sub>	x

DECISION TABLE 11.A.9 (15)

Data Requirement

h	x
w	x
F <sub>y</sub>	x

Web in Axial Compression	Y	N	N	I	
Web in Flexural Compression		N	Y	N	I
Web in Combine Axial and Flexural Compression		N	N	Y	I
h/w > 12000/F <sub>y</sub>		N	N	N	Y
Execute Table 11.A.9	Y				
Execute Table 11.A.10		Y			
Execute Table 11.A.11			Y		
MSG: Max h/w of the web exceeded				Y	

$h/w \leq 255/\sqrt{F_y}$	Y	N	h	w	F <sub>y</sub>	
$h/w > 255/\sqrt{F_y}$		N	Y	h	w	F <sub>y</sub>
Web Class = 1	Y					
Web Class = 4		Y				



DECISION TABLE 11.A.10 (16)

Data Requirement

h	x	
w	x	
F <sub>y</sub>	x	

$\frac{h}{w} \leq \frac{420}{\sqrt{F_y}}$ $\frac{420}{\sqrt{F_y}} < \frac{h}{w} \leq \frac{520}{\sqrt{F_y}}$ $\frac{520}{\sqrt{F_y}} < \frac{h}{w} \leq \frac{590}{\sqrt{F_y}}$ $\frac{h}{w} > \frac{690}{\sqrt{F_y}}$	Y N N N N Y N N N N Y N N N N Y	h w F <sub>y</sub> h w F <sub>y</sub> h w F <sub>y</sub> h w F <sub>y</sub>
Web Class = 1 Web Class = 2 Web Class = 3 Web Class = 4	Y Y Y Y	

DECISION TABLE 11.A.11 (17)

Data Requirement

$\frac{P_f}{C_y}$	x	Table 7.2.2 (93)
$\frac{P_f}{C_y} > 0.15$	Y N	$P_f C_y$
Execute Table 11.A.12 Execute Table 11.A.13	Y Y	



DECISION TABLE 11.A.12 (18)

Data Requirement

<div>h</div> <div>w</div> <div>F<sub>y</sub></div> <div>P<sub>f</sub></div> <div>C<sub>y</sub></div>	x	<div>Table 7.2.2 (93)</div>
	x	
	x	
	x	
	x	

DECISION TABLE 11.A.13 (19)

Data Requirement

<div>h</div> <div>w</div> <div>F<sub>y</sub></div> <div>P<sub>f</sub></div> <div>C<sub>y</sub></div>	x	<div>Table 7.2.2 (93)</div>
	x	
	x	
	x	
	x	

<div><math display="block">h/w \leq 420/\sqrt{F_y} (1-1.4 \frac{P_f}{C_y})</math><math display="block">420/\sqrt{F_y} (1-1.4 \frac{P_f}{C_y}) &lt; \frac{h}{w} \leq 450/\sqrt{F_y} (1-0.43 \frac{P_f}{C_y})</math><math display="block">h/w &gt; 450/\sqrt{F_y} (1-0.43 \frac{P_f}{C_y})</math></div>	<div>Y N N</div> <div>N Y N</div> <div>N N Y</div>	<div>h w F<sub>y</sub> P<sub>f</sub> C<sub>y</sub></div> <div>h w F<sub>y</sub> P<sub>f</sub> C<sub>y</sub></div> <div>h w F<sub>y</sub> P<sub>f</sub> C<sub>y</sub></div>
<div>Web Class = 1</div> <div>Web Class = 2</div> <div>Web Class = 4</div>	<div>Y</div> <div>Y</div> <div>Y</div>	

<div><math display="block">h/w \leq 420/\sqrt{F_y} (1-1.4 \frac{P_f}{C_y})</math><math display="block">420/\sqrt{F_y} (1-1.4 \frac{P_f}{C_y}) &lt; \frac{h}{w} \leq 520/\sqrt{F_y} (1-1.28 \frac{P_f}{C_y})</math><math display="block">520/\sqrt{F_y} (1-1.28 \frac{P_f}{C_y}) &lt; \frac{h}{w} \leq 690/\sqrt{F_y} (1-2.6 \frac{P_f}{C_y})</math><math display="block">h/w &gt; 690/\sqrt{F_y} (1-2.6 \frac{P_f}{C_y})</math></div>	<div>Y N N N</div> <div>N Y N N</div> <div>N N Y N</div> <div>N N N Y</div>	<div>h w F<sub>y</sub> P<sub>f</sub> C<sub>y</sub></div> <div>h w F<sub>y</sub> P<sub>f</sub> C<sub>y</sub></div> <div>h w F<sub>y</sub> P<sub>f</sub> C<sub>y</sub></div> <div>h w F<sub>y</sub> P<sub>f</sub> C<sub>y</sub></div>
<div>Web Class = 1</div> <div>Web Class = 2</div> <div>Web Class = 3</div> <div>Web Class = 4</div>	<div>Y</div> <div>Y</div> <div>Y</div> <div>Y</div>	



DECISION TABLE 11.8 (20)

## Data Requirement

T-Section	x
I-Section	x
J-Section	x
Rectangular Hollow Section	x
Box Section	x

DECISION TABLE 11.8.1 (21)

## Data Requirement

T-Section	x
I-Section	x
Double Angle	x
b	x
t	x
$F_y$	x

T-Section	Y
I-Section	N
J-Section	N
Rectangular Hollow Section	N
Box Section	N
Execute Table 11.8.1	Y
Execute Table 11.8.2	Y
Execute Table 11.8.3	Y
Execute Table 11.8.4	Y

T-Section	Y
I-Section	N
Double Angle	N
$b/t \leq 54/\sqrt{F_y}$	N
$54/\sqrt{F_y} < b/t \leq 64/\sqrt{F_y}$	Y
$64/\sqrt{F_y} < b/t \leq 100/\sqrt{F_y}$	N
$b/t > 100/\sqrt{F_y}$	Y
Flange Class = 1	Y
Flange Class = 2	Y
Flange Class = 3	Y
Flange Class = 4	Y
Execute Table 11.8.5	Y
Class = 3	Y
Class = 4	Y





DECISION TABLE 11.B.3 (23)

Data Requirement

b	x	
t	x	
F <sub>y</sub>	x	

$b/t \leq 160/\sqrt{F_y}$	Y N N N	b t F <sub>y</sub>
$160/\sqrt{F_y} < b/t \leq 200/\sqrt{F_y}$	N Y N N	b t F <sub>y</sub>
$200/\sqrt{F_y} < b/t \leq 255/\sqrt{F_y}$	N N Y N	b t F <sub>y</sub>
$b/t > 255/\sqrt{F_y}$	N N N Y	b t F <sub>y</sub>
Flange Class = 1	Y	
Flange Class = 2	Y	
Flange Class = 3	Y	
Flange Class = 4	Y	

DECISION TABLE 11.B.2 (22)

Data Requirement

b	x	
t	x	
F <sub>y</sub>	x	

$b/t \leq 100/\sqrt{F_y}$	Y N	b t F <sub>y</sub>
$b/t > 100/\sqrt{F_y}$	N Y	b t F <sub>y</sub>
Flange Class = 3	Y	
Flange Class = 4	Y	



DECISION TABLE 11.B.4 (24)

Data Requirement

b	x
t	x
F <sub>y</sub>	x

$b/t \leq 200/\sqrt{F_y}$	Y N N	b t F <sub>y</sub>
$200/\sqrt{F_y} < b/t \leq 255/\sqrt{F_y}$	N Y N	b t F <sub>y</sub>
$b/t > 255/\sqrt{F_y}$	N N Y	b t F <sub>y</sub>
Flange Class = 1	Y	
Flange Class = 3	Y	
Flange Class = 4	Y	







DECISION TABLE X.4 (27)

Data Requirement

R <sub>1</sub> R <sub>2</sub> R <sub>3</sub>	Table 13.2 (28) Table X.5 (42) Table X.5 (42)
--	---

$R_1 + R_2 + R_3 \leq 1.0$	Y N	R <sub>1</sub> R <sub>2</sub> R <sub>3</sub>
MSG: Strength Criterion Satisfied MSG: Strength Criterion Not Satisfied	Y Y	

DECISION TABLE 12.3.A.1 (26)

Data Requirement

S <sub>K</sub> (K = 1 to 4) g <sub>K</sub> (K = 1 to 4)	x x
--	--------

Condition	I	
REDUTM = $\sum_K \frac{S_K^2}{4g_K}$	Y	S(1) S(2) S(3) S(4) g(1) g(2) g(3) g(4)





DECISION TABLE 13.2 (28)

## Data Requirement

$P_f > 0$	x
$P_f < 0$	x
$L_x$	x
$L_y$	x
$r_x$	x
$r_y$	x
TSLRAX	
TSLRAY	

Table 13.2 (28)  
Table 13.2 (28)

DECISION TABLE 13.2.A.1 (29)

## Data Requirement

$A_n$	x	Table 12.3
$A_g$	x	
$F_y$	x	
$F_u$	x	
$\phi$	x	Table 7.2.2
$P_f$	x	
Member Using Pin Connections		
$P_{rt}$	x	Table 13.2.A.1

$P_f > 0$ (Comp)	Y	Y	N	N
$P_f < 0$ (Tension)	N	Y	N	Y
Execute Table 13.3.1	Y			
$TSLRAX = L_x/r_x +$				$L_x r_x$
$TSLRAY = L_y/r_y +$				$L_y r_y$
$TLR = \max(TSLRAX, TSLRAY)$				TSLRAX TSLRAY
Execute Table 13.2.A.1				
$R_1 = 0$			Y	

+ In subroutine, check if SLRAX, SLRAY > 300.0.  
If yes, print out message and stop execution of program.

Member Using Pin-Connections $A_n/A_g \geq F_y/F_u$	I I Y Y N I	$A_n A_g F_y F_u$
$P_{rt} = \min(\phi A_n F_y, 0.85 \phi A_n F_u)$	Y	$\phi A_n F_y F_u$
$P_{rt} = \min[\phi(F_u/A_g) A_n, 0.85 \phi A_n F_u]$	Y	$\phi A_n A_g F_u$
$P_{rt} = 0.75 \phi A_n F_y$	Y	$\phi A_n F_y$
$F_1 = \frac{P_f}{P_{rt}}$	Y Y Y	$P_f P_{rt}$



DECISION TABLE 13.3.1 (30)

## Data Requirement

Class = 1	Table 11.(5)
Class = 2	Table 11.(5)
Class = 3	Table 11.(5)
Class = 4	Table 11.(5)
UKLR	Table 9.3 (4)
$F_y$	$\lambda$
$E$	$x$

Class = 1	Y N N N	UKLR $F_y E$
Class = 2	N Y N N	
Class = 3	N N Y N	
Class = 4	N N N Y	
$\lambda = UKLR \sqrt{F_y / \pi^2 E}$	Y Y Y Y	
Execute Table 13.3.1.A.1	Y Y Y	
MSG: This section is a Class 4 section. Checking should use Clause 12 of CSA S136. "Cold Formed Steel Structural Members". Any other messages produced from now on relating to design should be ignored.	Y	

DECISION TABLE 13.3.1.A.1 (31)

## Data Requirement

$\lambda$	Table 13.3.1 (30)
$\phi$	
$A_g$	$x$
$F_y$	$x$
$P_f$	$x$
$P_{rc}$	
	Table 7.2.2 (93)
	Table 13.3.1.A.1

$0 \leq \lambda \leq 1.0$	Y N N N	$\lambda$
$1.0 < \lambda \leq 2.0$	N Y N N	$\lambda$
$2.0 < \lambda \leq 3.6$	N N Y N	$\lambda$
$3.6 < \lambda$	N N N Y	$\lambda$
$P_{rc} = \phi A_g F_y (1.035 - 0.201 \lambda - 0.224 \lambda^2)$	Y	$\phi A_g F_y \lambda$
$P_{rc} = \phi A_g F_y (-0.111 + 0.63 \lambda^{-1} + 0.094 \lambda^{-2})$	Y	$\phi A_g F_y \lambda$
$P_{rc} = \phi A_g F_y (0.012 + 0.867 \lambda^{-2})$	Y	$\phi A_g F_y \lambda$
$P_{rc} = \phi A_g F_y \lambda^{-2}$	Y	$\phi A_g F_y \lambda$
$R_1 = \frac{P_f}{P_{rc}}$	Y Y Y Y	$P_f P_{rc}$



DECISION TABLE 11.B.5 (32)

Data Requirements

Is the Double Angle Continuously Connected by Adequate Mechanical Fasteners or Welds	X
---	---

Is the Double Angle Continuously Connected by Adequate Mechanical Fasteners or Welds	Y N
Execute Table 11.A.2 Class = 3	Y Y



## DECISION TABLE 13.5.A (43)

## Data Requirement

Class = 1			Table 11. (5)
Class = 2			Table 11. (5)
Class = 3			Table 11. (5)
Class = 4			Table 11. (5)
$\phi$		x	
Z		x	
F <sub>y</sub>		x	
S		x	
M <sub>fx</sub>			Table 7.2.2 (93)
M <sub>rx</sub>			Table 13.6.A.1 (47)

Class = 1	Y	N	N	N
Class = 2	N	Y	N	N
Class = 3	N	N	Y	N
Class = 4	N	N	N	Y
$M_{rx} = \phi z F_y = (\phi M_p)$	Y	Y		$\phi z F_y$
$M_{rx} = \phi s F_y = (\phi M_y)$		Y		$\phi s F_y$
Execute Table 13.5.A.1			Y	
$R_2 = M_{fx}/M_{rx}$	Y	Y	Y	$M_{fx} M_{rx}$
$R_3 = 0$	Y	Y	Y	

## DECISION TABLE X.5 (42)

## Data Requirement

Continuous Laterally Supported Member	x
Laterally Unsupported Member	x

Continuous Laterally Supported Member	Y	N
Laterally Unsupported Member	N	Y
$M_{fy} = 0$	Y	
Execute Table 13.5.A	Y	
Execute Table 13.6.A	Y	





DECISION TABLE 13.5.A.1 (44)

Data Requirement

Flange Class = 1	Table 11.8 (20)
Flange Class = 2	Table 11.8 (20)
Flange Class = 3	Table 11.8 (20)
Flange Class = 4	Table 11.8 (20)
Web Class = 1	Table 11.A (6)
Web Class = 2	Table 11.A (6)
Web Class = 3	Table 11.A (6)
Web Class = 4	Table 11.A (6)
h	
w	x
F <sub>y</sub>	x
F <sub>cr</sub>	x
φ	
s	x
A <sub>w</sub>	x
A <sub>f</sub>	x
M <sub>fx</sub>	x
M <sub>rx</sub>	x
	Table 13.5.3 (45)
	Table 7.2.2 (93)
	Table 13.6.A.1 (47)



DECISION TABLE 13.5.A.1 (44)

Flange Class = 1 Flange Class = 2 Flange Class = 3 Flange Class = 4 Web Class = 1 Web Class = 2 Web Class = 3 Web Class = 4  $12000/F_y \geq \frac{h}{w} > 690/\sqrt{F_{cr}}$  $\frac{h}{w} < 690/\sqrt{F_{cr}}$	Y N Y N N N N N N N N Y N N Y N N N N N N N Y N N Y N N N N N N N N N Y Y Y Y Y N N N N N Y N N N N N N N N N N Y N N N N N N N N N N Y N N Y Y Y Y Y N N N Y  Y Y Y N N N I I I  N N N Y Y I I I I	h w F <sub>y</sub> F <sub>cr</sub>  h w F <sub>cr</sub>
$M_{rx} = \text{Min} (\phi S F_{cr}, \phi S F_y)$ $M_{rx} = \text{Min} (\phi S F_y, \phi S F_{cr}) [1 - 0.0005 \frac{A_w}{A_f} (\frac{h}{w} - 690/\sqrt{F_{cr}})]$ MSG: $M_{rx}$ to be determined from Clause 12, CSA S136  $R_2 = M_{rx}/M_{rx}$	Y Y Y Y Y Y  Y Y Y  Y  Y Y Y Y Y Y Y Y	$\phi S F_{cr} F_y$  $\phi S F_{cr} A_w A_f h w F_y$    $M_{rx} M_{rx}$



## DECISION TABLE 13.6.A (46)

## Data Requirements

## DECISION TABLE 13.5.3 (45)

## Data Requirement

b	x
t	x
F <sub>y</sub>	x
K <sub>b</sub>	x

M <sub>rx</sub>	x	Table 13.6.A.1 (47)
M <sub>fx</sub> > 0	x	
M <sub>fy</sub> > 0	x	
M <sub>fx</sub> and M <sub>fy</sub> > 0	x	
Doubly Symmetrical Hollow Section (Circular Hollow/ Square Hollow)	x	

$b/t \leq 201 \sqrt{K_b/F_y}$	Y N	K <sub>b</sub> F <sub>y</sub> b t
$F_{cr} = F_y [1.46 - 0.004 \sqrt{F_y/K_b} (b/t)]$	Y	F <sub>y</sub> K <sub>b</sub> b t
$F_{cr} = 26200 K_b / (b/t)^2$	Y	K <sub>b</sub> b t

Doubly Symmetrical Hollow Section (Circular Hollow/ Square Hollow)	N N N Y Y	N N N Y Y
M <sub>fx</sub> > 0	Y N Y Y N Y	Y N Y Y N Y
M <sub>fy</sub> > 0	N Y Y N Y Y	N Y Y N Y Y
M <sub>fx</sub> and M <sub>fy</sub> > 0	N N Y N N Y	N N Y N N Y
M <sub>ry</sub> = M <sub>rx</sub> Execute Table 13.6.A.1 Execute Table 13.6.A.2	Y Y Y Y Y	M <sub>rx</sub> Y Y



DECISION TABLE 13.6.A.1 (47)

## Data Requirement

Double Sym. Section Channel Prevented from Twisting	x
Class = 1	x
Class = 2	x
Class = 3	x
Class = 4	x

Table 11. (5)  
Table 11. (5)  
Table 11. (5)  
Table 11. (5)

Double Sym. Section Channel Prevented From Twisting	Y	Y	Y	N	N	N	N	N	N
Class = 1	N	N	N	Y	Y	Y	Y	N	N
Class = 2	Y	N	N	Y	N	N	N	I	I
Class = 3	N	Y	N	N	N	Y	N	N	I
Class = 4	N	N	Y	N	N	N	Y	N	I

Execute Table 13.6.1  
Execute Table 13.6.2  
MSG: Section Not a Double  
Sym. Section  
A Rational Method of  
Analysis such as given  
in the Column Research  
Council's "Guild to Design  
Criteria for Compression  
Members" should be used to  
Calculate Resistant Moments.

DECISION TABLE 13.6.1 (48)

## Data Requirement

$M_u$	x	Table 13.6.2.A (51)
$M_p$	x	
$\phi$	x	
z	x	
$F_y$	x	
$M_{fx}$	x	
$M_{rx}$	x	

Table 7.2.2 (93)  
Table 13.6.A.1 (47)

$M_u > 2/3 M_p$	Y	N	$M_u$	$M_p$
$M_{rx} = \text{Min}[\phi F_y, 1.15\phi M_p(1 - \frac{0.28 M_p}{M_u})]$	Y		$\phi z F_y$	$M_p$
$M_{rx} = \phi M_u$		Y	$\phi M_u$	
$R_2 = M_{fx}/M_{rx}$	Y	Y	$M_{fx}$	$M_{rx}$





DECISION TABLE 13.6.2 (49)

Data Requirement

Class = 3	Table 11. (5)
Class = 4	Table 11. (5)
$M_u$	x
$M_y$	x
$\phi$	x
s	x
$F_y$	x
$F_{cr}$	x
$M_{fx}$	x
$M_{rx}$	x

$M_u > \frac{2}{3} M_y$	Y Y N N	$M_u M_y$
Class = 3	Y N Y N	
Class = 4	N Y N Y	
$M_{rx} = \min[\phi s F_y, 1.15 \phi M_y (1 - \frac{0.28 M_y}{M_u})]$	Y	$\phi s F_y M_y M_u$
$M_{rx} = \phi M_u$	Y	$\phi M_u$
$M_{rx} = \min[\phi s F_{cr}, 1.15 \phi M_y (1 - \frac{0.28 M_y}{M_u})]$	Y	$\phi s F_{cr} M_y M_u$
$M_{rx} = \phi M_u$	Y	$\phi M_u$
$R_2 = M_{fx} / M_{rx}$	Y Y Y Y	$M_{fx} M_{rx}$

DECISION TABLE 13.6.A.2 (50)

Data Requirement

Class = 1	Table 11. (5)
Class = 2	Table 11. (5)
Class = 3	Table 11. (5)
Class = 4	Table 11. (5)
$\phi$	x
$z_y$	x
$F_y$	x
$s_y$	x
$F_{cr}$	x
$M_{fy}$	
$M_{ry}$	

Class = 1	Y N N N	
Class = 2	N Y N N	
Class = 3	N N Y N	
Class = 4	N N N Y	
$M_{ry} = \phi z_y F_y (= \phi M_p)$	Y Y	$\phi z_y F_y$
$M_{ry} = \phi s_y F_y (= \phi M_y)$	Y	$\phi s_y F_y$
$M_{ry} = \phi s_y F_{cr}$	Y	$\phi s_y F_{cr}$
$R_3 = M_{fy} / M_{ry}$	Y Y Y Y	$M_{fy} M_{ry}$



DECISION TABLE 13.6.2.A (51)

Data Requirement

I-Section	x	Table 13.6.2.A.1
$w_x$	x	
L	x	
E	x	
$I_y$	x	
G	x	
J	x	
$C_w$	x	
S	x	
$\sigma_1$	x	Table 13.6.1.A.1
$\sigma_2$	x	Table 13.6.1.A.1

I-Section	Y	N
$M_U = \frac{\pi}{\omega_x L} \sqrt{E I_y G J + \frac{\pi^2}{L^2} E I_y C_w}$	Y	$\omega_x L E I_y G J C_w$
$M_U = \frac{S}{\omega_x} \sqrt{\sigma_1^2 + \sigma_2^2}$	Y	$S \omega_x \sigma_1 \sigma_2$

DECISION TABLE 13.6.2.A.1 (52)

Data Requirement

Beam with Both Ends With Effective Lateral Support For the Compression Flange	x
Beam with Only One End with Effective Lateral Support For the Compression Flange	x

Beam with Both Ends With Effective Lateral Support For the Compression Flange	Y	N
Beam with Only One End with Effective Lateral Support for the Compression Flange	N	Y
Execute Table 13.8.3.a	Y	
$\omega_x = 1.0$	Y	



DECISION TABLE 13.8.3.a (53)

## Data Requirement

Member not Subjected to Transverse Loads in the Minor Axis Plane Between Supports	x
Member Subjected to Distributed Load or Series of Point Loads in the Minor Axis Plane Between Supports	x
Member Subjected to a Concentrated Load or Moment in the Minor Axis Plane Between Supports	x

DECISION TABLE 13.8.3.a.A.1 (54)

## Data Requirement

Sway Effects About the Major Axis Included in Analysis	x
Member Bent in Single Curvature About the Major Axis	x
Member Bent in Double Curvature About the Major Axis	x
$M_{f1}$	x
$M_{f2}$	x

Member Not Subjected to Transverse Loads in Minor Axis Plane Between Supports	y	n	n
Member Subjected to Distributed Load or Series of Point Loads in Minor Axis Plane Between Supports	n	y	n
Member Subjected to a Concentrated Load or Moment in Minor Axis Plane Between Supports	n	n	y
Execute Table 13.8.3.a.A.1	y	y	y
$\omega_x = 1.0$			
$\omega_x = 0.85$			

Sway Effects About the Major Axis Included in Analysis	n	n	y
Member Bent in Single Curvature About the Major Axis	y	n	y
Member Bent in Double Curvature About the Major Axis	n	y	n
$\omega_x = 0.6 + 0.4 M_{f1}/M_{f2}$	y		
$\omega_x = 0.6 - 0.4 M_{f1}/M_{f2}$	y		
$\omega_x = 1.0$	y		
$\omega_x = 0.85$	y		
$M_{f1}$			$M_{f1}$
$M_{f2}$			$M_{f2}$



DECISION TABLE 13.6.1.A.1 (55)

Data Requirement

$A_f$	x
$L$	x
$d$	x
$r_t$	x

Condition	I	
$\sigma_1 = 20000 A_f/Ld$	Y	$A_f L d$
$\sigma_2 = 250000 r_t^2/L^2$	Y	$r_t L$

DECISION TABLE X.6 (56)

Data Requirement

Axial Compression and Bending	x
Axial Tension and Bending	x

Axial Compression and Bending	Y
Axial Tension and Bending	N
Execute Table 13.8	Y
Execute Table 13.9	Y





DECISION TABLE 13.8 (57)

Data Requirement

Double Symmetrical Hollow Section (Circular Hollow/Square Hollow)	x
I-Section	x
Class = 1	Table 11. (5)
Class = 2	Table 11. (5)
Class = 3	Table 11. (5)
Class = 4	Table 11. (5)
M <sub>rx2</sub>	Table 13.8.A.3 (62)

Double Symmetrical Hollow Section (Circular Hollow/Square Hollow)	N N N N N N N N	Y
I-Section	Y Y Y N N N N N	Y
Class = 1	Y N N N Y N N N	M <sub>rx2</sub>
Class = 2	N Y N N N Y N N	M <sub>rx2</sub>
Class = 3	N N Y N N N Y N	Y
Class = 4	N N N Y N N N Y	Y Y Y Y Y Y
Execute Table 13.8.2	Y Y	
M <sub>rx1</sub> = M <sub>rx2</sub>		
M <sub>ry1</sub> = M <sub>rx2</sub>		
Execute Table 13.8.1		



DECISION TABLE 13.8.2 (58)

$\frac{M_{fx}}{M_{rx1}} + \frac{M_{fy}}{M_{ry1}} \leq 1.0$	Y Y Y N N N N Y	$M_{fx} M_{rx1} M_{fy} M_{ry1}$
$\frac{P_f}{P_{rc2}} + \frac{0.85 M_{fx}}{M_{rx1}} + \frac{0.6 M_{fy}}{M_{ry1}} \leq 1.0$	N Y Y Y N N Y N	$P_f P_{rc2} M_{fx} M_{rx1} M_{fy} M_{ry1}$
$\frac{P_f}{P_{rc1}} + \frac{\omega_x M_{fx}}{M_{rx2}(1 - \frac{P_f}{C_{ex}})} + \frac{\omega_y M_{fy}}{M_{ry1}(1 - \frac{P_f}{C_{ey}})} \leq 1.0$	N N Y Y Y N N Y	$P_f P_{rc1} M_{fx} \omega_x$ $M_{rx2} C_{ex} M_{ry1}$ $C_{ey} \omega_y$
MSG: Both Strength and Stability Criteria Not Satisfied MSG: Strength Satisfied; Stability Not Satisfied MSG: Strength and Stability Criteria Satisfied MSG: Stability Satisfied; Strength Not Satisfied	Y Y Y Y Y Y	

DECISION TABLE 13.8.2 (58)

Data Requirement

$M_{fx}$	Table 7.2.2 (93)
$M_{rx1}$	Table 13.8.A.1 (60)
$M_{fy}$	Table 7.2.2 (93)
$M_{ry1}$	Table 13.8.A.1 (60)
$P_f$	Table 7.2.2 (93)
$P_{rc2}$	Table 13.8.A.8 (67)
$P_{rc1}$	Table 13.8.A.6 (65)
$\omega_x$	Table 13.6.2.A.1 (52)
$M_{rx2}$	Table 13.8.A.3 (62)
$C_{ex}$	Table 13.8.A.9 (68)
$C_{ey}$	Table 13.8.A.9 (68)
$\omega_y$	Table 13.8.A.10 (69)



DECISION TABLE 13.8.1 (59)

Data Requirement

P <sub>f</sub>	Table 7.2.2 (93)
P <sub>rc2</sub>	Table 13.8.A.8 (67)
M <sub>fx</sub>	Table 7.2.2 (93)
M <sub>rx1</sub>	Table 13.8.A.1 (60)
M <sub>fy</sub>	Table 7.2.2 (93)
M <sub>ry1</sub>	Table 13.8.A.1 (60)
P <sub>rc1</sub>	Table 13.8.A.6 (65)
ω <sub>x</sub>	Table 13.6.2.A.1(52)
M <sub>rx2</sub>	Table 13.8.A.3 (62)
C <sub>ex</sub>	Table 13.8.A.9 (68)
ω <sub>y</sub>	Table 13.8.A.10(69)
C <sub>ey</sub>	Table 13.8.A.9 (68)



DECISION TABLE 13.8.1 (59)

$\frac{P_f}{P_{rc2}} + \frac{M_{fx}}{M_{rx1}} + \frac{M_{fy}}{M_{ry1}} \leq 1.0$	Y Y N N	$P_f \quad P_{rc2} \quad M_{fx} \quad M_{rx1} \quad M_{fy} \quad M_{ry1}$
$\frac{P_f}{P_{rc1}} + \frac{\omega_x M_{fx}}{M_{rx2}(1-C_{ex})} + \frac{\omega_y M_{fy}}{M_{ry1}(1-C_{ey})} \leq 1.0$	Y N Y N	$P_f \quad P_{rc1} \quad \omega_x \quad M_{fx} \quad M_{rx2} \quad C_{ex}$ $\omega_y \quad M_{fy} \quad M_{ry1} \quad C_{ey}$
MSG: Both Strength and Stability Criteria Satisfied	Y	
MSG: Strength Satisfied; Stability not Satisfied Design Unsatisfactory	Y	
MSG: Strength not Satisfied; Stability Satisfied Design Unsatisfactory	Y	
MSG: Both Strength and Stability Criteria Not Satisfied Design Unsatisfactory	Y	





DECISION TABLE 13.8.A.1 (60)

Data Requirement

Class = 1		Table 11. (5)
Class = 2		Table 11. (5)
Class = 3		Table 11. (5)
Class = 4		Table 11. (5)
$\phi$	x	
$M_p$	x	
s	x	
Fy	x	
zy	x	
sy	x	

DECISION TABLE 13.8.A.2 (61)

Data Requirement

Flange Class = 3		Table 11.8 (20)
Flange Class = 4		Table 11.8 (20)
Fy	x	
h	x	
w	x	
Fcr		Table 13.5.3 (45)
$\phi$	x	
s	x	
Aw	x	
Af	x	
sy	x	

Flange Class = 3		Y Y N N N	
Flange Class = 4		N N Y Y Y	
$690/\sqrt{F_y} < h/w \leq 690/\sqrt{\phi_s F_{cr}/\phi_s}$	$h/w \leq 690/\sqrt{F_y}$	N N Y N N	Fy h w
$690/\phi_s F_{cr}/\phi_s < h/w \leq 12000/F_y$		Y N N Y N	Fy Fcr h w
		N Y N N Y	Fy Fcr h w
$M_{rx1} = \phi_s F_{cr}$		Y Y	$\phi_s F_{cr}$
$M_{rx1} = \phi_s F_{cr} [1 - 0.005 \frac{A_w h}{A_f w} - 690/\phi_s F_{cr}/\phi_s]$		Y	$\phi_s F_{cr} A_w A_f h w$
MSG: $M_{rx1}$ to be Determined by Clause 12, CSA S136			
$M_{ry1} = \phi_s F_{cr}$		Y Y	$\phi_s F_{cr}$
$M_{ry1} = \phi_s F_{cr} [1 - 0.005 \frac{A_w h}{A_f w} - 690/\phi_s F_{cr}/\phi_s]$		Y	$\phi_s F_{cr}$

Class = 1	Y N N N	
Class = 2	N Y N N	
Class = 3	N N Y N	
Class = 4	N N N Y	
$M_{rx1} = \phi M_p$	Y Y	$\phi M_p$
$M_{rx1} = \phi_s F_y$	Y	$\phi_s F_y$
$M_{ry1} = \phi_z F_y$	Y Y	$\phi_z F_y$
$M_{ry1} = \phi_s F_y$	Y	$\phi_s F_y$
Execute Table 13.8.A.2	Y	



DECISION TABLE 13.8.A.3 (62)

## Data Requirement

Double Symmetrical Section Channel Prevented from Twisting Class = 1 Class = 2 Class = 3 Class = 4	x x	Table 11. (5) Table 11. (5) Table 11. (5) Table 11. (5)
---	-----	--

DECISION TABLE 13.8.A.4 (63)

## Data Requirement

$M_u$ $M_p$ $\phi$	x x	Table 13.6.2.A (51)
--------------------------	--------	---------------------

Double Symmetrical Section Channel Prevented from Twisting Class = 1 Class = 2 Class = 3 Class = 4	Y Y Y Y N N N N N N N N N Y Y N N N Y N N N Y Y N N I N Y N N N Y N N I N N Y N N N Y N I N N N Y N N N Y I	Y Y Y Y Y Y Y Y
Execute Table 13.8.A.4 Execute Table 13.8.A.5 MSG: Section Not a Doubly Symmetrical Section A Rational Method of Analysis Such as Given in CRC "Guide to Design Criteria for Metal Comp. Members" Should be Used. To Calculate Resistance Moment.	Y Y Y Y Y Y Y Y	Y Y Y Y

$M_u > 2/3 M_p$		Y N	$M_p M_u$
$M_{rx2} = \text{Min}[\phi M_p, 1.15\phi M_p(1 - \frac{0.28M_p}{M_u})]$		Y	$\phi M_p M_u$
$M_{rx2} = \phi M_u$		Y	$\phi M_u$



DECISION TABLE 13.8.A.5 (64)

Data Requirement

$M_u$ $M_y$ $\phi$ $S$ $F_y$ $F_{cr}$	$x$ $x$ $x$ $x$ $x$	Table 13.6.2.A (51)
		Table 13.5.3 (45)

$M_u > \frac{2}{3} M_y$ Class = 3 Class = 4	$Y$ $Y$ $N$ $Y$ $N$ $Y$ $Y$	$M_y M_u$
$M_{rx2} = \text{Min}[\phi S F_y, 1.15 \phi M_y (1 - \frac{0.28 M_y}{M_u})]$	$Y$	$\phi S F_y M_y M_u$
$M_{rx2} = \phi M_u$	$Y$	$\phi M_u$
$M_{rx2} = \text{Min}[\phi S F_{cr}, 1.15 \phi M_y (1 - \frac{0.28 M_y}{M_u})]$	$Y$	$\phi S F_{cr} M_y M_u$

DECISION TABLE 13.8.A.6 (65)

Data Requirement

$\lambda$ $\phi$ $A_g$ $F_y$	$x$ $x$ $x$ $x$	Table 13.8.A.7 (66)
---------------------------------------	--------------------------	---------------------

$0 \leq \lambda \leq 1.0$ $1.0 < \lambda \leq 2.0$ $2.0 < \lambda \leq 3.6$ $3.6 < \lambda$	$Y$ $N$ $N$ $N$ $Y$ $N$ $N$ $Y$	$\lambda$ $\lambda$ $\lambda$ $\lambda$
$P_{rc1} = \phi A_g F_y (1.035 - 0.201 \lambda + 0.224 \lambda^2)$	$Y$	$\phi A_g F_y \lambda$
$P_{rc1} = \phi A_g F_y (-0.111 + 0.63 \lambda^{-1} + 0.094 \lambda)^{-2}$	$Y$	$\phi A_g F_y \lambda$
$P_{rc1} = \phi A_g F_y (0.012 + 0.867 \lambda)^{-2}$	$Y$	$\phi A_g F_y \lambda$
$P_{rc1} = \phi A_g F_y \lambda^{-2}$	$Y$	$\phi A_g F_y \lambda$



DECISION TABLE 13.8.A.7 (66)

Data Requirement

UKLR $F_y$ $E$	x x	Table 9.3 (4)
----------------------	--------	---------------

DECISION TABLE 13.8.A.8 (67)

Data Requirement

$\phi$ $A_g$ $F_y$	x x x	
--------------------------	-------------	--

Condition	I	
$\lambda_1 = UKLR/F_y/\eta^2 E$	Y	UKLR $F_y$ E

Condition	I	
$P_{rc2} = \phi A_g F_y$	Y	$\phi A_g F_y$





DECISION TABLE 13.8.A.9 (68)

Data Requirement

$A_g$	x	Table 9.3 (4)
$K_x$	x	
$L_x$	x	
$r_x$	x	Table 9.3 (4)
$K_y$	x	
$L_y$	x	
$r_y$	x	

Condition	I	
$C_{ex} = 286000 A_g / (K_x L_x / r_x)^2$	Y	$A_g K_x L_x r_x$
$C_{ey} = 286000 A_g / (K_y L_y / r_y)^2$	Y	$A_g K_y L_y r_y$

DECISION TABLE 13.8.A.10 (69)

Data Requirement

Beam With Both Ends With Effective Lateral Support For the Compression Flange	x
Beam With Only One End With Effective Lateral Support For the Compression Flange	x

Beam With Both Ends With Effective Lateral Support For the Compression Flange	Y	N
Beam With Only One End With Effective Lateral Support For the Compression Flange	N	Y
Execute Table 13.8.A.11 $w_y = 1.0$	Y	Y



DECISION TABLE 13.8.A.11 (70)

## Data Requirement

Member Not Subjected To Transverse Loads in the Major Axis Plane Between Supports	x
Member Subjected to Distributed Load or Series of Point Loads In the Major Axis Plane Between Supports.	x
Member Subjected to a Concentrated Load or Moment in the Major Axis Plane Between Supports	x

DECISION TABLE 13.8.A.12 (71)

## Data Requirement

Sway Effects About the Minor Axis Included in Analysis/Resisted by Bracing or Shear Wall	x
Member Bent in Single Curvature About the Minor Axis	x
Member Bent in Double Curvature About the Minor Axis	x
$M_{fly}$	x
$M_{f2y}$	x

Member Not Subjected to Transverse Loads in the Major Axis Plane Between Supports	Y N N
Member Subjected to Distributed Load or Series of Point Loads in the Major Axis Plane Between Supports	N Y N
Member Subjected to a Concentrated Load or Moment in the Major Axis Plane Between Supports	N N Y
Execute Table 13.8.A.12 $\omega_y = 1.0$ $\omega_y = 0.85$	Y Y Y

Sway Effects About the Minor Axis Included in Analysis/Resisted by Bracing or Shear Wall	N N Y
Member Bent in Single Curvature About the Minor Axis.	Y N Y N
Member Bent in Double Curvature About the Minor Axis.	N Y N Y
$\omega_y = 0.6 + 0.4 M_{fly}/M_{f2y}$ $\omega_y = 0.6 - 0.4 M_{fly}/M_{f2y}$ $\omega_y = 1.0$ $\omega_y = 0.85$	Y Y Y Y
	$M_{fly}$ $M_{f2y}$ $M_{fly}$ $M_{f2y}$



DECISION TABLE 13.9 (72)

## Data Requirement

$P_f$	Table 7.2.2 (93)
$P_{rt}$	Table 13.2.A.1 (29)
$M_{fx}$	Table 7.2.2(93)
$M_{r1}$	Table 13.9.A.1 (73)
$M_{fy}$	Table 7.2.2 (93)
$M_{rx2}$	Table 13.8.A.3 (62)
$M_{ry1}$	Table 13.8.A.1 (60)

$\frac{P_f}{P_{rt}} + \frac{M_{fx}}{M_{r1}} + \frac{M_{fy}}{M_{r1}} \leq 1.0$	Y N Y N	$P_f$ $P_{rt}$ $M_{fx}$ $M_{r1}$ $M_{fy}$
$\frac{M_{fx}}{M_{rx2}} + \frac{M_{fy}}{M_{ry1}} - \frac{P_f}{P_{rt}} \leq 1.0$	Y N N Y	$M_{fx}$ $M_{rx2}$ $M_{fy}$ $M_{ry1}$ $P_f$ $P_{rt}$
Tensile and Compressive Stress Criteria Satisfied	Y	
Tensile Stress Criterion Not Satisfied	Y	
Compressive Stress Criterion Not Satisfied	Y	
Tensile and Compressive Stress Criteria Not Satisfied	Y	

DECISION TABLE 13.9.A.1 (73)

## Data Requirement

Class = 1		Table 11. (5)
Class = 2		Table 11. (5)
Class = 3		Table 11. (5)
Class = 4		Table 11. (5)

Class = 1	Y	
Class = 2	Y	
Class = 3	Y	
Class = 4	Y	
$M_{r1} = \phi M_p$	Y Y	$\phi M_p$
$M_{r1} = \phi M_y$	Y Y	$\phi M_y$



DECISION TABLE 13.4 (74)

Data Requirement

$V_f$ $V_r$	Table 7.2.2 (93) Table 13.4.A.1 (75)
----------------	---

$\frac{V_f}{V_r} \leq 1.0$	Y N	$V_f$ $V_r$
Shear Criterion Satisfied	Y	
Shear Criterion Not Satisfied	Y	

DECISION TABLE 13.4.A.1 (75)

Data Requirement

Elastic Analysis	x	Table 13.4.1 (76)
Plastic Analysis	x	
Flexural Member Subjected to Shear	x	
Pin	x	
$A_w$	x	
$F_s$	x	
$\phi$	x	
$A_g$	x	
$F_y$	x	
$w$	x	
$d$	x	
Gusset Plate	x	

Elastic Analysis	Y Y Y N	
Plastic Analysis	N N N Y	
Flexural Member Subjected to Shear	Y N N Y	
Gusset Plate	N Y N I	
Pin	N N Y I	
$V_r = \phi A_w F_s$	Y	
$V_r = 0.5 \phi A_g F_y$	Y	
$V_r = 0.66 \phi A_g F_y$	Y	
$V_r = 0.55 \phi w d F_y$	Y	
	$A_w F_s$ $\phi A_g F_y$ $\phi A_g F_y$ $\phi w d F_y$	





DECISION TABLE 13.4.1 (76)

Data Requirement

h	x	Table 13.4.1.A.1 (77) Table 13.4.1.A.1 (77) Table 13.4.1.A.1 (77)
w	x	
K <sub>v</sub>		
F <sub>y</sub>	x	
T		
η		

$\frac{h}{w} \leq 167\sqrt{K_v/F_y}$	Y N N N	$h w K_v F_y$
$167\sqrt{K_v/F_y} < \frac{h}{w} \leq 190\sqrt{K_v/F_y}$	N Y N N	$h w K_v F_y$
$190\sqrt{K_v/F_y} < \frac{h}{w} \leq 239\sqrt{K_v/F_y}$	N N Y N	$h w K_v F_y$
$239\sqrt{K_v/F_y} < \frac{h}{w}$	N N N Y	$h w K_v F_y$
$F_s = 0.66 F_y$	Y	$F_y$
$F_s = 110\sqrt{F_y K_v}/(h/w)$	Y	$F_y K_v h$
$F_s = 110\sqrt{F_y K_v} \cdot T/(h/w) + \eta F_y$	Y	$F_y K_v T h w \eta$
$F_s = 26200 K_v \cdot T/(h/w)^2 + \eta F_y$	Y	$K_v T h w \eta F_y$

DECISION TABLE 13.4.1.A.1 (77)

Data Requirement

Stiffened Webs	x
a	x
h	x

Stiffened Webs	N Y Y	a h
a/h < 1.0	I Y N	a h
a/h ≥ 1.0	I N Y	a h
$K_v = 5.34$	Y	
T = 1.0	Y	
η = 0.0	Y	
$K_v = 4 + 5.34/(a/h)^2$	Y	a h
$T = 1 - 0.866/\sqrt{1 + (a/h)^2}$	Y Y	a h
$\eta = 0.5/\sqrt{1 + (a/h)^2}$	Y Y	a h
$K_v = 5.34 + 4/(a/h)^2$	Y	a h



DECISION TABLE 8.5 (78) PLASTIC ANALYSIS

Web Stiffener Are Supplied On The Member At A Point Of Load Application Where A Plastic Hinge Would Form	Y N Y Y Y
Splices In The Member Are Designed To Transmit 1.1 Times The Max. Computed Moment Under Factored Loads At The Splice Location Or 0.25 M <sub>p</sub> Whichever Is Greater	Y I N Y Y
Member Is Not Subjected To Repeated Heavy Impact Or Fatigue	Y I I N Y
The Influence Of Inelastic Deformation On The Strength Of The Structure Shall Be Taken Into Account	Y I I I N
Execute Table 8.5a MSG: Limitations On Plastic Analysis Not Satisfied. Check Clause 8.5(d,e,f,g)	Y Y Y Y Y

DECISION TABLE 8.5 (78)

Data Requirement

Web Stiffener Supplied On The Member At A Point Of Load Application Where A Plastic Hinge Would Form	x
Splices In the Member Are Designed To Transmit 1.1 Times The Maximum Computed Moment Under Factored Loads At The Splice Location Or 0.25 M <sub>p</sub> Whichever Is Greater	x
Member Is Not Subjected To Repeated Heavy Impact Or Fatigue	x
The Influence Of Inelastic Deformation On The Strength Of The Structure Shall Be Taken Into Account	x



DECISION TABLE 8.5.a (79)

Data Requirement

$F_y$	x	
$F_u$	x	

$F_y \leq 0.8 F_u$	Y N	$F_y F_u$
Execute Table 8.5.b MSG: Limitations on Plastic Analysis Not Satisfied, Check Clause 8.5(a)	Y	Y

DECISION TABLE 8.5.b (80)

Data Requirement

Class = 1	Table 11. (5)
-----------	---------------

Class = 1	Y N
Execute Table 8.5.c MSG: Limitations on Plastic Analysis Not Satisfied. Check Clause 8.5(b)	Y



DECISION TABLE 13.7 (82)

Data Requirement

$M_{f1}$	x	
$M_{f2}$	x	
$r_y$	x	
$F_y$	x	

DECISION TABLE 8.5.c (81)

Data Requirement

$L_{ph}$ $L_{cr}$	x	Table 13.7 (82)
----------------------	---	-----------------

Laterally Unbraced Length Between Plastic Hinges $\leq L_{cr}$	Y N	$L_{ph} L_{cr}$
Execute Table X.2	Y	
MSG: Limitations on Plastic Analysis Not Satisfied Check Clause 13.7	Y	

$\frac{M_{f1}}{M_{f2}} > 0.5$	Y N	$M_{f1} M_{f2}$
$\frac{M_{f1}}{M_{f2}} \leq 0.5$	N Y	$M_{f1} M_{f2}$
$L_{cr} = 210 r_y / \sqrt{F_y}$	Y	$r_y F_y$
$L_{cr} = 375 r_y / \sqrt{F_y}$	Y	$r_y F_y$





DECISION TABLE 15. (83)

Data Requirement

Cover Plate Used	x
Cover Plate Used	Y N
Execute Table 15.A.1	Y
Execute Table X.3	Y

DECISION TABLE 15.A.1 (84)

Data Requirement

Bolted Girder	x
Welded Girder	x
A <sub>cov</sub>	x
A <sub>f</sub>	x
P <sub>cov</sub>	x
M <sub>fc</sub>	x
Y	x
I <sub>cov</sub>	x

Bolted Girder Welded Girder $A_{cov} \leq 70\% A_f$ $P_{cov} \geq \frac{A_{cov} M_{fc}}{I_{cov}}$	Y Y Y Y N N N N N Y Y Y N N I  Y N Y N I	$A_{cov} A_f$ $P_{cov} A_{cov} M_{fc} Y I_{cov}$
MSG: Clause 15.4.2 and 15.4.4 Satisfied MSG: Clause 15.4.4 Not Satisfied MSG: Clause 15.4.2 Not Satisfied Execute Table X.3	Y Y Y Y Y Y	



DECISION TABLE 15.5 (85)

Data Requirement

Interior Bearing Stiffener	x
End Bearing Stiffener	x
$T_{web}$	x
$A_{web}$	
$A_{bst}$	x

Table 15.5 (85)

DECISION TABLE 15.5.A.1 (86)

Data Requirement

$K_x L_x$	x
-----------	---

Table 9.3 (4)

$K_x L_x > 3/4 L_x$	Y	N	$K_x L_x$
MSG: Clause 15.5.2, KL of Stiffener Satisfied	Y		
MSG: $K_x L_x$ of Stiffener Does Not Satisfy Clause 15.5.2	Y		
Execute Table X.3	Y		

Interior Bearing Stiffener	Y	N
End Bearing Stiffener	N	Y
$A_{web} = T_{web} \times 25$	Y	$T_{web}$
$A_{web} = T_{web} \times 12$	Y	$T_{web}$
$A_g = A_{web} + A_{bst}$	Y	$A_{web}$
Execute Table 15.5.A.1	Y	$A_{bst}$







DECISION TABLE 15.6.A.1 (88)

Data Requirement

Table 13.4.1.A.1 (77)	
K <sub>y</sub>	x
F <sub>y</sub>	x
h	x
w	x
F <sub>y</sub> stif	x

Condition	I
$C = \text{MAX}(1.0 - \frac{45000.0 K_y}{F_y (h/w)^2}, 0.1)$	Y
YRAT = F <sub>y</sub> /F <sub>y</sub> stif	Y
	K <sub>y</sub> F <sub>y</sub> h w
	F <sub>y</sub> F <sub>y</sub> stif

DECISION TABLE 15.6.A.2 (89)

Data Requirement

Intermediate Stiffener Furnished in Pairs	x
Intermediate Single Angle Stiffener	x
Intermediate Single Plate Stiffener	x

Intermediate Stiffener Furnished in Pairs	Y	N	N
Intermediate Single Angle Stiffener	N	Y	N
Intermediate Single Plate Stiffener	N	N	Y
SFACT = 1.0	Y		
SFACT = 1.8		Y	
SFACT = 2.4			Y





DECISION TABLE 15.6.A.3. (90)

Data Requirement

$V_{fadj}$ $V_r$	x	Table 13.4.A.1 (75)
---------------------	---	---------------------

If $\frac{V_{fadj}}{V_r} < 1.0$	Y N	$V_{fadj}$ $V_r$
REFAC = $\frac{W_{fadj}}{W_r}$	Y	$V_{fadj}$ $V_r$
REFAC = 1.0	Y	

DECISION TABLE 13.9 (91)

Data Requirement

$B_{r1}$ SLOAD	x	Table 13.9.A.1 (92)
-------------------	---	---------------------

$B_{r1} > \text{SLOAD}$	Y N	$B_{r1}$ SLOAD
MSG: Clause 15.9 (Stability of Thin Webs) Satisfied MSG: Clause 15.9 Not Satisfied	Y Y	



DECISION TABLE 7.2.2 (93)

Data Requirement

$P_f = 0$	x	Table 7.2.5 (96)
$M_{fx} = 0$	x	Table 7.2.3 (94)
$M_{fy} = 0$	x	Table 7.2.3 (94)
$V_f = 0$	x	Table 7.2.3 (94)
$B_f = 0$	x	Table 7.2.3 (94)
$\gamma$		
$\alpha_D$		
$\alpha_L$		
$\alpha_Q$		
$\alpha_T$		
$P_D$	x	
$P_L$	x	
$P_Q$	x	
$P_T$	x	
$M_{XD}$	x	
$M_{XL}$	x	
$M_{XQ}$	x	
$M_{XT}$	x	
$M_{YD}$	x	
$M_{YL}$	x	
$M_{YQ}$	x	
$M_{YT}$	x	
$V_D$	x	
$V_L$	x	
$V_Q$	x	
$V_T$	x	
$B_D$	x	
$B_L$	x	
$B_Q$	x	
$B_T$	x	
$\psi$	x	Table 7.2.4 (95)

DECISION TABLE 13.9.A.1 (92)

Data Requirement

Flange Is Restrained Against Rotation	x
Flange Is Not Restrained Against Rotation	x
$\phi$	x
$h$	x
$w$	x
$a$	x
ATW	x

Flange Is Restrained Against Rotation	Y N
Flange Is Not Restrained Against Rotation	N Y

$B_{r1} = \phi \frac{16700}{(h/w)^2} [5.5 + \frac{4}{(a/h)^2}] \times \text{ATW}$	Y	$\phi h w a \text{ ATW}$
$B_{r1} = \phi \frac{16700}{(h/w)^2} [2 + \frac{4}{(a/h)^2}] \times \text{ATW}$	Y	$\phi h w a \text{ ATW}$

NOTE: For Distributed Load  
 ATW = Panel Length x Web Thickness  
 For Concentrated Loads and  
 Loads Distributed Partially  
 ATW = Less of a or h x Web Thickness







DECISION TABLE 7.2.3 (94)

Data Requirement

Overturning, Uplift Or Stress Reversal Case	x
--	---

DECISION TABLE 7.2.4 (95)

Data Requirement

When One of L, T, Q Act	x
When Two of L, T, Q Act	x
When All of L, T, Q Act	x

Overturning, Uplift Or Stress Reversal Case	Y N
$\alpha_H = 1.25$ $\alpha_D = 0.85$ $\alpha_L = 1.5$ $\alpha_Q = 1.5$ $\alpha_T = 1.25$	Y Y Y Y Y Y Y Y

When One of L, T, Q Act	Y N N
When Two of L, T, Q Act	N Y N
When All of L, T, Q Act	N N Y
$\psi = 1.0$ $\psi = 0.7$ $\psi = 0.6$	Y Y Y





DECISION TABLE 7.2.5 (96)

Data Requirement

<p>Farm Building of Low Occupancy Rate, Density &lt; One Person/500 sq. ft. During Normal Periods of Use of 4 hrs. or Longer/ Buildings For Which It Can Be Shown That Collapse Is Not Likely To Cause Injury</p>	<p>X</p>
<p>Farm Building of Low Occupancy Rate, Density &lt; One Person/500 sq. ft. During Normal Periods of Use of 4 hrs. or Longer/ Buildings For Which It Can Be Shown That Collapse Is Not Likely To Cause Injury.</p>	<p>N Y</p>
<p>Y = 1.0 Y = 0.8</p>	<p>Y Y</p>

DECISION TABLE 13.10 (97)

Data Requirement

R <sub>B</sub>	Table 13.10.A.1 (98)		
R <sub>B</sub> ≤ 1.0	Y	N	R <sub>B</sub>
	MSG: Clause 13.10 Bearing Resistance Satisfied MSG: Clause 13.10 Bearing Resistance Not Satisfied		



DECISION TABLE 13.10.A.1 (98)

Data Requirement

On Contact Area Of Machined, Accurately Sawn Or Fitted Parts	x	Table 7.2.2 (93) Table 13.10.A.1 (98)
On Expansion Rollers Or Rockers	x	
$\phi$	x	
$F_y$	x	
Acontact	x	
D	x	
L	x	
$B_f$	x	
$B_r$	x	

On Contact Area of Machined, Accurately Sawn or Fitted Parts	Y N	
On Expansion Rollers or Rockers	N Y	
$B_r = 1.5\phi F_y A_{contact}$	Y	$\phi F_y A_{contact}$
$B_r = 0.0008\phi D L F_y^2$	Y	$\phi D L F_y$
$R_B = \frac{B_f}{B_r}$	Y Y	$B_f B_r$



DECISION TABLE X.6.R (56R)

Data Requirement

Axial Compression And Bending Axial Tension And Bending	x x
Axial Compression And Bending Axial Tension And Bending	Y N N Y
CHECKI=0.0 Execute Table 13.8.R Execute Table 13.9	Y Y Y Y

DECISION TABLE 13.8.R (57R)

Data Requirement

Double Symmetric Hollow Section (Circular Hollow/Square Hollow)	x	Table 11. (5) Table 11. (5) Table 11. (5) Table 11. (5)
I-Section Class = 1 Class = 2 Class = 3 Class = 4	x	
Double Sym. Hollow Section (Circular Hollow/Square Hollow)		
I-Section Class = 1 Class = 2 Class = 3 Class = 4	N N N N N N N Y Y Y Y N N N N N Y N N N Y N N N I N Y N N N Y N N I N N Y N N N Y N I N N N Y N N N Y I	
CHECKN=2.0 CHECKN=3.0 CHECKI=CHECKI+1.0 Execute Table 13.8.2.R $M_{rx1} = M_{rx2}$ $M_{ry1} = M_{ry2}$ Execute Table 13.8.1.R CHECKI Eq. CHECKN?	Y Y	$M_{rx2}$ $M_{ry2}$









DECISION TABLE 13.8.1.R (59R)

## Data Requirement

CHECKI	Table 13.8 (56R)
P <sub>f</sub>	Table 7.2.2 (93)
M <sub>fx</sub>	Table 7.2.2 (93)
M <sub>fy</sub>	Table 7.2.2 (93)
M <sub>rxA</sub>	Table X.7 (99R)
M <sub>ryA</sub>	Table X.7 (99R)
P <sub>rCA</sub>	Table X.7 (99R)
ω <sub>xA</sub>	Table X.7 (99R)
ω <sub>yA</sub>	Table X.7 (99R)
θ <sub>x</sub>	Table X.7 (99R)
θ <sub>y</sub>	Table X.7 (99R)

DECISION TABLE 13.8.1.R (59R)

CHECKI=1 CHECKI=2	Y N Y N N Y N Y	CHECKI CHECKI
$\frac{P_f}{P_{rCA}} + \frac{M_{fx}\omega_{xA}}{M_{rxA}\theta_x} + \frac{M_{fy}\omega_{yA}}{M_{ryA}\theta_y} \leq 1.0$	Y Y N N	P <sub>f</sub> M <sub>fx</sub> M <sub>fy</sub> M <sub>rxA</sub> M <sub>ryA</sub> ω <sub>xA</sub> ω <sub>yA</sub> θ <sub>x</sub> θ <sub>y</sub>
MSG: Strength Criterion Satisfied	Y	
MSG: Stability Criterion Satisfied	Y	
MSG: Strength Criterion Not Satisfied	Y	
MSG: Stability Criterion Not Satisfied	Y	



DECISION TABLE X.7.R (99R)

Data Requirement

CHECKN	Table 13.8.R (57R)
CHECKI	Table X.6.R (56R)
M <sub>rx1</sub>	Table 13.8.A.1 (60)
M <sub>rx2</sub>	Table 13.8.A.3 (62)
M <sub>ry1</sub>	Table 13.8.A.1 (60)
P <sub>rc1</sub>	Table 13.8.A.6 (65)
P <sub>rc2</sub>	Table 13.8.A.8 (67)
ω <sub>x</sub>	Table 13.6.2.A.1 (52)
ω <sub>y</sub>	Table 13.8.A.10 (69)
P <sub>f</sub>	Table 7.2.2 (93)
C <sub>ex</sub>	Table 13.8.A.9 (68)
C <sub>ey</sub>	Table 13.8.A.9 (68)

DECISION TABLE X.7.R (99R)

CHECKN=2.0	Y Y N N N	CHECKN
CHECKN=3.0	N N Y Y Y	CHECKN
CHECKI=1.0	Y N Y N N	CHECKI
CHECKI=2.0	N Y N Y N	CHECKI
CHECKI=3.0	I I N N Y	CHECKI
M <sub>rxA</sub> =M <sub>rx1</sub>	Y Y Y	M <sub>rx1</sub>
M <sub>rxA</sub> =M <sub>rx2</sub>	Y Y Y	M <sub>rx2</sub>
M <sub>ryA</sub> =M <sub>ry1</sub>	Y Y Y Y Y	M <sub>ry1</sub>
P <sub>rcA</sub> =P <sub>rc1</sub>	Y Y Y	P <sub>rc1</sub>
P <sub>rcA</sub> =P <sub>rc2</sub>	Y Y Y	P <sub>rc2</sub>
ω <sub>xA</sub> =1.0	Y Y	ω <sub>x</sub>
ω <sub>xA</sub> =0.85	Y Y	ω <sub>x</sub>
ω <sub>xA</sub> =ω <sub>x</sub>	Y Y	ω <sub>x</sub>
ω <sub>yA</sub> =1.0	Y Y	ω <sub>y</sub>
ω <sub>yA</sub> =0.6	Y Y	ω <sub>y</sub>
ω <sub>yA</sub> =ω <sub>y</sub>	Y Y	ω <sub>y</sub>
θ <sub>x</sub> =1.0	Y Y	θ <sub>x</sub>
θ <sub>x</sub> =(1-P <sub>f</sub> /C <sub>ex</sub> )	Y Y	P <sub>f</sub> C <sub>ex</sub>
θ <sub>y</sub> =1.0	Y Y	θ <sub>y</sub>
θ <sub>y</sub> =(1-P <sub>f</sub> /C <sub>ey</sub> )	Y Y	P <sub>f</sub> C <sub>ey</sub>



## APPENDIX C

### USER'S GUIDE

#### Introduction

This appendix discusses the processing program limitations, Input/Output procedures for interactive and batch mode and contains a description of MTS files. The source programs are listed in Appendices D and E.

#### C.1 Program Limitations

##### (a) Interactive Mode

Maximum number of tables	= 120
Maximum number of rules per table	= 40
Maximum number of ingredients per condition or action	= 11
Maximum number of data elements	= 700

##### (b) Batch Mode

Maximum number of tables	= 120
Maximum number of rules per table	= 40
Maximum number of ingredients per condition or action	= 9
Maximum number of data elements	= 700



## C.2 Batch Mode Control Cards

- (1) \$SIGNON XXXXXX P= T= PRIO=

R  
H  
N  
L
- (2) PASSWORD
- (3) \$RUN \*FORTG SCARDS=BATCHMODE+ROUTINE1+ROUTINE2+  
ROUTINE3+ROUTINE4+ROUTINE5+  
ROUTINE6+ROUTINE7+ROUTINE8
- (4) \$RUN -LOAD# 8=CSAS16
- (5) DATA DECK
- (6) \$SIGNOFF

See Sect. C.5.5 for description of source and object files in item 3.

See Sect. C.5 for MTS I/O Units in item 4.

See Sect. C.3 for contents of data deck in item 5.

## C.3 Data Input For Batch Mode

### (a) Control Cards:

- (1) 

11

14

THEMAP

 } FORMAT(T11,A4)

THEMAP Prints out the contents of the permanent and temporary arrays.

= YES or NO





(2) 

10	ISAVE	
----	-------	--

FORMAT(I10)

ISAVE=1 indicates decision table information has to be input

ISAVE=2 indicates decision table information has been processed from previous runs, and the following items 3 to 8 will not be necessary. This information storage capability, on average, saves 25 seconds of CPU time per run.

(b) Decision Table Input: Items (3) to (5) to be supplied for each decision table.

(3) 

10	20	30	40	
T	LT	MT	NT	

FORMAT (4I10)

T = decision table number

LT = number of rules

MT = number of actions

NT = number of conditions

(4) 

1	5	11	50	51	79	80
INDEX		ENTRIES	INGREDIENTS			FLAG

FORMAT(T51,5(I4,1X),I4,A1,T1,I5,5X,40I1)



INDEX = condition or action stub subscript

(condition stub: condition data subscripts)

(action stub: action data subscripts; or

blank if action is a message;

or a table number if action is

to execute a table)

ENTRIES = condition or action entries

(condition entry: 1=YES 2=NO)

(action entry: 1= calculating value of  
an element or printing  
message

2= execution of another  
table)

INGREDIENTS = the data subscripts of condition or  
action ingredients

FLAG = 'c', if more ingredients on the next card

= 'blank' if no more ingredients on the next  
card.

(5) Read this card if FLAG=c in 4

51                      79



FORMAT(T51,5(I4,1X),I4)

(6)

BLANK CARD

this indicates the end of decision table input



- (c) Data Element Properties: Item (7) to be supplied for each data element referenced by decision tables

(7) 

10	20	30
KGLOB	TABDK	NSET

 } FORMAT(3I10)

KGLOB = data subscript

TABDK = table number of KGLOB

NSET = mutually exclusive set number of KGLOB

(8) 

BLANK CARD
------------

This indicates the end of the data element information

- (d) Problem Data: Values of data elements required to define problem

(9) 

5	11	20
KGLOB		DATAK

 } FORMAT(I5,5X,F10.0)

KGLOB = external input data subscript

DATAK = corresponding data value.

(10) 

BLANK CARD
------------

This indicates the end of the external input data

- (e) Instructions for Execution:

(11) 

10
TFIRST

 } FORMAT(I10)



TFIRST = the number of first decision table to be  
executed

11 14  
(12) 

	TRACE	
--	-------	--

 FORMAT(T11,A4)

TRACE = 'YES', if a trace of the tables executed for  
the problem is desired.

= 'NO', if no trace is desired.

11 14  
(13) 

	WARAY	
--	-------	--

 FORMAT(T11,A4)

WARAY = 'YES', if listing of compacted arrays is  
desired

= 'NO', if no listing is desired

A schematic arrangement of the batch-mode input  
cards is illustrated in Fig. C.1.

#### C.4 Interactive Mode Control Commands

\$signon XXXX

password

\$run obcombine 5=filename1 9=filename2 8=filename3

2=filename4 6=\*print\* 4=\*source\*

7=\*sink\*

For details on I/O units:

I/O unit 5 (see Sect. C.5.1)





I/O unit 2 (see Sect. C.5.3)

I/O unit 9 (see Sect. C.5.2)

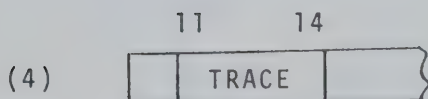
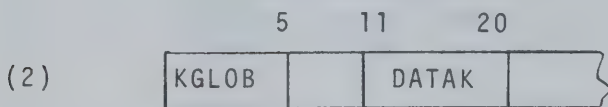
I/O unit 8,6,4,7 (see Sect. C.5.4)

The files associated with the I/O units are discussed in Sect. C.6. If decision table and data property information has already been processed and stored, the only data required to execute a problem are those associated with I/O units 5, and 2.

### C.5 I/O Units For Interactive Mode

(Note: For definition of variable names see Sect. C.3).

#### C.5.1 Data In I/O Unit 5





	11	14
(5)	WARAY	

### C.5.2 Data In I/O Unit 9

This file contains decision table information and data properties in coded form.

	10	20	30	40	
(1)	T	LT	MT	NT	(

	1	5	11	50	51	79	80
(2)	INDEX		ENTRIES	INGREDIENTS		FLAG	

	51	79
(3)		INGREDIENTS

(4) BLANK CARD

	10	20	30
(5)	KGLOB	TABDK	NSET

(6) BLANK CARD

### C.5.3 Data In I/O Unit 2

	11	14
(1)	THEMAP	



10

(2)



#### C.5.4 Data In I/O Units 8, 6, 4, 7

- I/O Unit 8 - The file which this unit refers to contains the processed decision table information in compacted binary form.
- I/O Unit 6 - 6 = \*print\* refers to output for the line printer.
- I/O Unit 4 - 4 = \*SOURCE\* refers to input from the terminal.
- I/O Unit 7 - 7=\*sink\* refers to output for the terminal.

#### C.6 Description Of MTS Files

- (a) OBCOMBINE - this file is an MTS control file. The contents of which is listed in Fig. C.2. This file is in object form.
- (b) OBCOMBINER - this file is used with the recursive execution scheme. It's contents and function are similar to that of OBCOMBINE. A listing of this file is presented in Fig. C.3.
- (c) DECIDATA1 - this file contains coded but unprocessed decision tables and data



- property information. It is referenced by I/O unit 9 during execution.
- (d) DECIDATA1R - this file is used with the recursive execution scheme. Its content and function are similar to that of DECIDATA1.
- (e) CSAS16 - this file contains processed decision tables and data property information in binary form. It is referenced by I/O unit 8 during execution.
- (f) CSAS16R - this file is used with the recursive execution scheme. Its content and function are similar to that of CSAS16.
- (g) MAPNSAVE - this file contains the values of variables THEMAP and ISAVE (Sect. C.5.3).
- (h) BATCHMODE - this file contains the subroutines SPECHK, SETUP, INITIAL, SETS, OUTPUT and STAK in source form for batch mode execution.
- (i) LUB - this file contains the external input data for example 1 in Sect. 8.1. A listing of this file is presented in Fig. C.4. The file is referenced by I/O unit 5.





- (j) AXCOM - this file contains the externally input data for example 2 in Sect. 8.2. A listing of this file is presented in Fig. C.5. The file is referenced by I/O unit 5.
- (k) COMBEN - this file contains the externally input data for example 3 in Sect. 8.3. A listing of this file is presented in Fig. C.6. The file is referenced by I/O unit 5.
- (l) ROUTINE1 TO ROUTINE8 - these files contain the condition and action subroutines of the decision tables compiled in Appendix B.



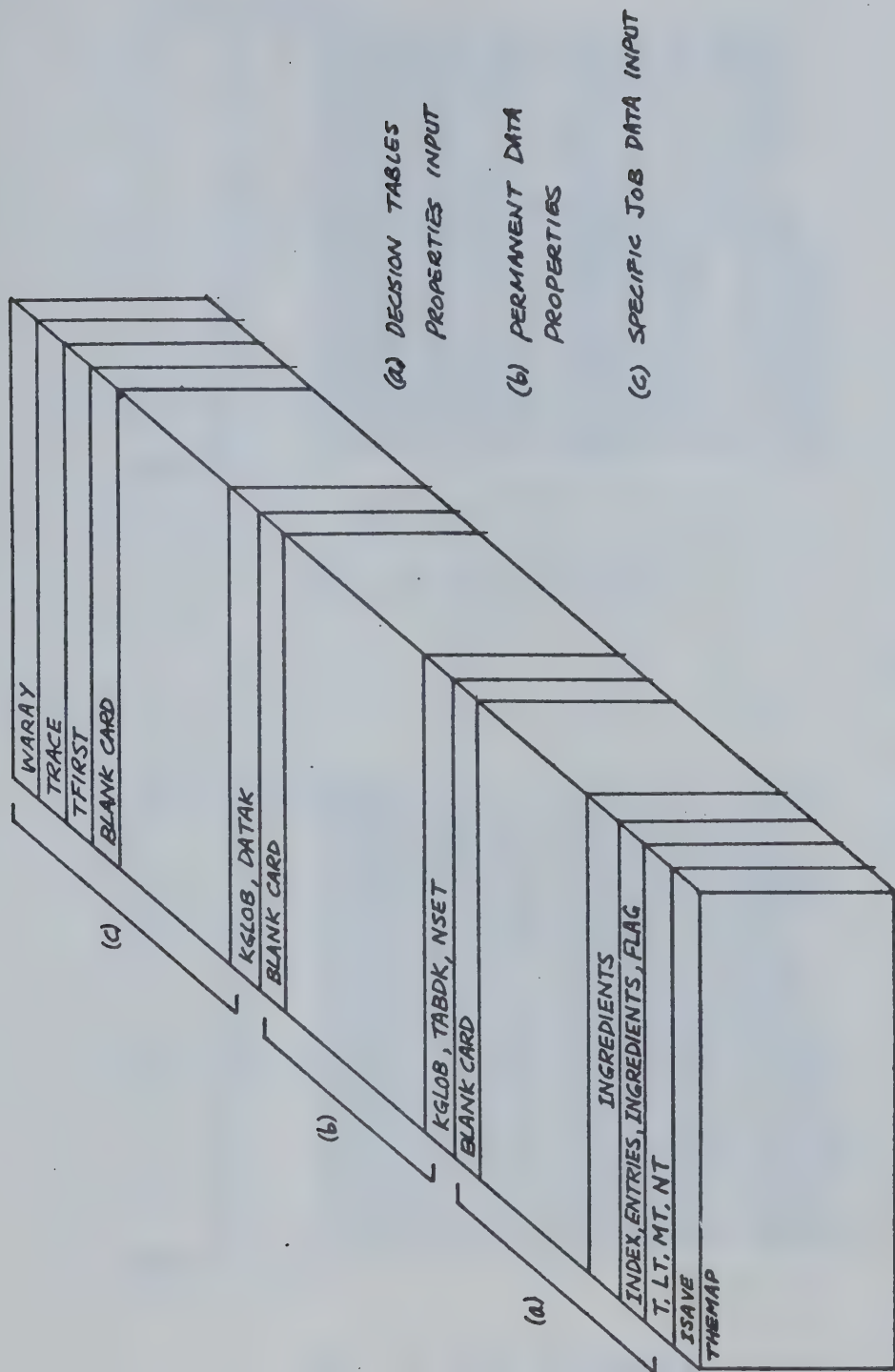


FIG. C.1 ARRANGEMENT OF DATA INPUT CARDS FOR BATCH MODE



```

$!list obcombine
> 1 $CONTINUE WITH OBSPECHK RETURN
> 2 $CONTINUE WITH OBSTAK RETURN
> 3 $CONTINUE WITH OBSETUP RETURN
> 4 $CONTINUE WITH OBINITIAL RETURN
> 5 $CONTINUE WITH OBINPUT RETURN
> 6 $CONTINUE WITH OBSETS RETURN
> 7 $CONTINUE WITH OBOUTPUT RETURN
> 7.1 $CONTINUE WITH OBREADIN RETURN
> 8 $CONTINUE WITH OBROUTINE1 RETURN
> 9 $CONTINUE WITH OBROUTINE2 RETURN
> 10 $CONTINUE WITH OBROUTINE3 RETURN
> 11 $CONTINUE WITH OBROUTINE4 RETURN
> 12 $CONTINUE WITH OBROUTINE5 RETURN
> 13 $CONTINUE WITH OBROUTINE6 RETURN
> 14 $CONTINUE WITH OBROUTINE7 RETURN
> 15 $CONTINUE WITH OBROUTINE8 RETURN
#END OF FILE

```

FIG. C.2 FILE OBCOMBINE

```

$!list obcombiner
> 1 $CONTINUE WITH OBSPECHKR RETURN
> 2 $CONTINUE WITH OBSTAK RETURN
> 3 $CONTINUE WITH OBSETUP RETURN
> 4 $CONTINUE WITH OBINITIAL RETURN
> 5 $CONTINUE WITH OBINPUT RETURN
> 6 $CONTINUE WITH OBSETS RETURN
> 7 $CONTINUE WITH OBOUTPUT RETURN
> 8 $CONTINUE WITH OBREADIN RETURN
> 8.1 $CONTINUE WITH OBCLEAR RETURN
> 9 $CONTINUE WITH OBROUTINE1 RETURN
> 10 $CONTINUE WITH OBROUTINE2 RETURN
> 11 $CONTINUE WITH OBROUTINE3 RETURN
> 12 $CONTINUE WITH OBROUTINE4R RETURN
> 13 $CONTINUE WITH OBROUTINE5R RETURN
> 14 $CONTINUE WITH OBROUTINE6 RETURN
> 15 $CONTINUE WITH OBROUTINE7 RETURN
> 16 $CONTINUE WITH OBROUTINE8R RETURN
#END OF FILE

```

FIG. C.3 FILE OBCOMBINER



```

$!list lub
> 1,
> 1,
> 10,
> 22,
> 26,
> 29,
> 40,
> 47,
> 48,
> 58,
> 59,
> 64,
> 65,
> 76,
> 78,
> 79,
> 80,
> 119,
> 121,
> 122,
> 123,
> 124,
> 127,
> 128,
> 129,
> 130,
> 138,
> 169,
> 176,
>
> 1,
> 1,
> 10,
> 22,
> 26,
> 29,
> 40,
> 47,
> 48,
> 58,
> 59,
> 64,
> 65,
> 76,
> 78,
> 79,
> 80,
> 119,
> 121,
> 122,
> 123,
> 124,
> 127,
> 128,
> 129,
> 130,
> 138,
> 169,
> 176,
>
> 1.0,
> 1.0,
> 1.0,
> 1.0,
> 1.0,
> 1.0,
> 4.05,
> 5.18,
> 0.641,
> 4.04,
> 0.371,
> 12.19,
> 44.0,
> 29000.0,
> 72.5,
> 64.8,
> 1.0,
> 1.0,
> 0.0,
> 1.0,
> 0.0,
> 0.0,
> 1.0,
> 0.0,
> 0.0,
> 1.0,
> 3190.0,
> 1.0,
> 0.0,
>
> 177,
> 180,
> 187,
> 189,
> 197,
> 225,
> 320,
> 342,
> 334,
> 400,
> 401,
> 402,
> 403,
> 404,
> 405,
> 410,
> 411,
> 412,
> 413,
> 418,
> 419,
> 420,
> 421,
> 125,
> 0,
>
> 30
> 31
> 32
> 33
> 34
> 35
> 36
> 37
> 38
> 39
> 40
> 41
> 42
> 43
> 44
> 45
> 46
> 47
> 48
> 49
> 50
> 51
> 52
> 53
> 54
> 55
> 56 #END OF FILE
>
> 1.0,
> 0.0,
> 288.0,
> 2.2,
> 0.9,
> 1.0,
> 1.0,
> 1.0,
> 0.0,
> 0.0,
> 0.9,
> 1.25,
> 1.5,
> 1.5,
> 1.25,
> 1.0,
> 1036.8,
> 518.4,
> 0.0,
> 0.0,
> 14.4,
> 7.2,
> 0.0,
> 0.0,
> 0.0,
>
> YES
> YES

```

FIG. C.4 DATA FILE LUB





```

$!list axcom
>      1      1,
>      2      1,      1.0,
>      3      10,      1.0,
>      4      23,      1.0,
>      5      25,      0.0,
>      6      46,      16.7,
>      7      58,      0.63,
>      8      59,      8.0,
>      9      76,      44.0,
>     10      78,      29000.0,
>     11      119,      1.0,
>     12      121,      0.0,
>     13      122,      1.0,
>     14      123,      1.0,
>     15      124,      1.0,
>     16      125,      1.0,
>     17      166,      1.0,
>     18      179,      1.0,
>     19      186,      144.0,
>     20      188,      2.42,
>     21      197,      0.9,
>     22      208,      2.54,
>     23      216,      144.0,
>     24      400,      0.9,
>     25      401,      1.25,
>     26      402,      1.5,
>     27      403,      1.5,
>     28      404,      1.25,
>     29      405,      1.0,
>     30      406,      300.0,
>     31      407,      85.0,
>     32      408,      0.0,
>     33      409,      0.0,
>     34      220,      1.0,
>     35      221,      1.0,
>     36      0,
>     37      YES
>     38      YES
#END OF FILE

```

FIG. C.5 DATA FILE AXCOM







## APPENDIX D

### INTERACTIVE MODE PROCESSING PROGRAM

#### - SOURCE LISTING

This appendix presents a source listing of the Main routine and the subroutines of the processing program for interactive mode procedure. In order to conserve space, the condition and action subroutines of each decision table are not presented. An example of them is presented in Fig. 5.5.



## SLIST SPECHK

```

1  CC THIS IS THE MAIN SUBROUTINE DOING THE BULK OF DECISION TABLE
2  CC PROCESSING INCLUDING IDENTIFYING THE APPLICABLE RULE,
3  CC AND CHECKING THE PRESENCE OF DATA ETC.
4  C
5  CC DECLARATIONS
6  C
7  IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
8  INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDK,TFIRST,THPSET
9  COMMON /MICA/DATA,PRD
10 COMMON /MNSTUP/LARRY1,LARRY2,LARRY3,LARRY4,LARRY5,LARRY6,IBASE,
11 1 IPNTRC,IPNTRA,TABD,L,H,N,T,TFIRST
12 COMMON /MNSTIN/ISET,HEXSET,MARCA,TRACE,THENAP
13 COMMON/STOPIH/ICLEAR,IARROW
14 COMMON/STINTL/INDEX,ENTRY,INGR,IDEPND,THPSET
15 COMMON/MINCE/ICICLE
16 COMMON/DUNB/WARAY
17 DIMENSION
18 1 LARRY1(600),LARRY2(600),LARRY3(5000),LARRY4(5000),IBASE(120,4),
19 2 LARRY5( 600),LARRY6( 600),IPNTRC(600),IPNTRA(600),
20 3 DATA(700),PRD(700),TABD(700),ISET(700),
21 4L(120),H(120),N(120),STACK(20,5),RESULT(2)
22 DIMENSION HEXSET(150),MARCA(100),IARROW(700),ICLEAR(2000)
23 DATA YES/'YES'/,NO/'NO'/
24 C READ IF A MAP OF PERMANENT DATA STORAGE IS DESIRED OR NOT
25 READ(2,110) THENAP
26 110 FORMAT(T11,A4)
27 C IF ISAVE=1,ENTER SUBROUTINE SETUP,CALCULATE DECISION TABLES
28 C PERMANENT DATA,THEN STORE IN UNIT 8
29 C IF ISAVE=2,SKIP SUBROUTINE SETUP,READ DECISION TABLES
30 C PERMANENT DATA FROM UNIT 8
31 READ(2,100) ISAVE
32 100 FORMAT(I10)
33 CALL SETDSR(8,11000,11000)
34 GO TO (700,710),ISAVE
35 710 CALL INTIAL
36 READ(8) LARRY1,LARRY2,LARRY5,LARRY6,IPNTRC,IPNTRA
37 READ(8) LARRY3
38 READ(8) LARRY4
39 READ(8) ISET,TABD,L,H,N,T,TFIRST,IBASE
40 READ(8) HEXSET,MARCA,TRACE,ICLEAR,IARROW
41 C WHEN ISAVE=2
42 C READ TFIRST FROM UNIT 5 AGAIN IN CASE IT IS DIFFERENT
43 C THAN THAT READ FROM UNIT 8
44 READ(5,102) TFIRST
45 102 FORMAT(I10)
46 C IF WARAY EQUAL YES,WRITE THE ARRAYS IF STATMENT 900 TO 928
47 IF (WARAY.EQ.YES) GO TO 730
48 WRITE (6,900) (LARRY1(I),I=1,600)
49 900 FORMAT (1H1,5X,'LARRY1'//30(5X,10I5,5X,10I5/))
50 WRITE (6,902) (LARRY3(I),I=1,5000)
51 902 FORMAT (1H1,5X,'LARRY3'//50(5X,10(10I1,1X)/))
52 WRITE (6,904) (LARRY2(I),I=1,600)
53 904 FORMAT (1H1,5X,'LARRY2'//30(5X,10I5,5X,10I5/))
54 WRITE (6,906) (LARRY4(I),I=1,5000)
55 906 FORMAT (1H1,5X,'LARRY4'//50(5X,10(10I1,1X)/))
56 WRITE (6,908) (J,(IBASE(I,J),I=1,120),J=1,4)
57 908 FORMAT (1H1/4(5X,'IBASE',11//6(5X,10I5,5X,10I5/))//4))
58 WRITE(6,910) (LARRY5(I),I=1,600)
59 910 FORMAT (1H1,5X,'LARRY5'//30(5X,10I5,5X,10I5/))
60 WRITE (6,912) (LARRY6(I),I=1,600)
61 912 FORMAT (1H1,5X,'LARRY6'//30(5X,10I5,5X,10I5/))
62 WRITE (6,914) (IPNTRC(I),I=1,600)
63 914 FORMAT (1H1,5X,'IPNTRC'//30(5X,10I5,5X,10I5/))
64 WRITE (6,916) (IPNTRA(I),I=1,600)
65 916 FORMAT (1H1,5X,'IPNTRA'//30(5X,10I5,5X,10I5/))
66 WRITE (6,918) (IARROW(I),I=1,700)
67 918 FORMAT (1H1,5X,'IARROW'//35(5X,10I5,5X,10I5/))
68 WRITE (6,920) (ICLEAR(I),I=1,2000)
69 920 FORMAT (1H1,5X,'ICLEAR'//20(5(5X,10I5,5X,10I5/1/))
70 WRITE (6,922) (HEXSET(I),I=1,150)
71 922 FORMAT (1H1,20X,'HEXSET'//15(20X,10I5/))
72 WRITE (6,924) (MARCA(I),I=1,100)
73 924 FORMAT (1H0,20X'MARCA'//5(20X,10I5/))
74 WRITE(6,926) (ISET(I),I=1,700)

```





```

75      926 FORMAT(1H1,5X,'ISET'/35(5X,10I5,5X,10I5/))
76      WRITE(6,928) (L(I),I=1,120)
77      928 FORMAT(1H1,20X,'ARRAY L'/12(20X,10I5/))
78      GO TO 730
79      700 CALL SETUP
80      730 ICYCLE = 1
81      1 CALL INPUT(ICYCLE)
82      C PRINT DATA AGAIN TO CHECK ITS VALIDITY
83      WRITE(6,799)
84      799 FORMAT(1H1,15X,'DATA PRINTED AGAIN FOR CHECKING. ONLY ',
85      1 'THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE'//
86      2 31X,'KGLOBAL',10X,'DATAK',10X,'PRD'//)
87      DO 801 KGLOBAL = 1,700
88      IF (.NOT. PRD(KGLOBAL)) GO TO 801
89      WRITE(6,800) KGLOBAL,DATA(KGLOBAL),PRD(KGLOBAL)
90      800 FORMAT(25X,I10,1X,F14.4,5X,L7)
91      801 CONTINUE
92      C
93      CC GIVE A MESSAGE THAT EXECUTION OF CYCLE NUMBER (= ICYCLE) IS ABOUT
94      CC TO COMMENCE
95      C
96      WRITE(6,172) ICYCLE,TFIRST
97      172 FORMAT(1H1,10X,'CYCLE NUMBER',I3,5X,'*** START',
98      1 ' EXECUTION WITH TABLE',I4,' ***')
99      C INITIALISE BEFORE STARTING EXECUTION OF THE TABLES
100     TRUE = 1.0
101     T = TFIRST
102     ISTACK = 0
103     TABNO = 0
104     18 J = 1
105     19 I = 1
106     C GET THE ADDRESS OF THE CONDITION ENTRY
107     24 IJ = IBASE(T,3) + (J-1)*N(T) + I
108     C IF THE CONDITION ENTRY IS IMMATERIAL, SKIP CHECKING IT
109     IF (LARRY3(IJ) .EQ. 0) GO TO 52
110     25 I1 = IBASE(T,1) + I
111     KGLOBAL = LARRY1(I1)
112     C
113     CC FIRST CHECK IF THE CONDITION HAS BEEN SUPPLIED WITH ITS VALUE
114     C
115     2006 IF (PRD(KGLOBAL)) GO TO 50
116     C
117     CC CHECK IF ANY OTHER TABLE CAN BE EXECUTED TO GET THIS CONDITION
118     C
119     IF (TABD(KGLOBAL) .NE. 0) GO TO 45
120     C
121     CC OTHERWISE SEE IF THIS CONDITION CAN BE ESTABLISHED BY
122     CC SUBROUTINE CC OF THIS TABLE. THIS IS INDICATED BY
123     CC HAVING ATLEAST ONE INGREDIENT FOR THIS CONDITION
124     C
125     IF ((IPNTRC(I1+1) - IPNTRC(I1)) .NE. 0) GO TO 13
126     C
127     CC ELSE AN ERROR MESSAGE
128     C
129     12 WRITE(7,174) I,T,KGLOBAL
130     174 FORMAT(1H0,10X,'CONDITION NUMBER',I3,'OF TABLE',I3,
131     1 ' IS NOT AVAILABLE. THIS CORRESPONDS TO DATA NUMBER',I4,
132     2 11X,'SUBROUTINE READIN IS CALLED TO INPUT THIS DATA ITEM')
133     NUMBER=1
134     2000 KG=KGLOBAL
135     CALL READIN(KG,NUMBER,KGLOBAL)
136     IF (KG.NE.KGLOBAL) GO TO 2004
137     GO TO 2006
138     2004 WRITE(7,1000)
139     1000 FORMAT(1X,'YOU HAVE INPUT THE INCORRECT VALUE OF KG',
140     * ' PLEASE TRY AGAIN.GOOD LUCK')
141     GO TO 2000
142     C
143     CC CHECK IF ALL THE INGREDIENTS OF THIS CONDITION ARE PRESENT
144     C
145     13 IR = IPNTRC(I1) + 1
146     23 IDATA = LARRY5(IR)

```



```

147 2007 IF (.NOT. PRD(IDATA)) GO TO 39
148 15 IR = IR + 1
149 IF (IR .LE. IPNTRC(I1+1)) GO TO 23
150 C
151 CC A NORMAL EXIT FROM THIS LOOP INDICATES THAT ALL THE
152 CC DATA NECESSARY TO SET THIS CONDITION IS PRESENT AND SO
153 CC ITS SUBROUTINE CC CAN BE CALLED
154 C
155 GO TO (888,888,888,888,888,888,888,888,209,210,
156 * 211,212,213,214,215,216,217,218,219,888,
157 * 221,222,223,224,225,888,227,888,229,888,
158 * 231,888,888,888,888,888,888,888,888,888,
159 * 888,888,888,244,245,888,888,248,249,888,
160 5 888,888,888,888,888,888,888,258,259,888,
161 6 261,888,263,264,265,888,888,888,888,888,
162 7 888,272,888,274,888,276,277,888,279,888,
163 8 281,282,888,284,888,286,287,888,888,290,
164 9 291,888,888,888,888,888,888,297,888,888,888,
165 A 888,888,888,888,888,888,888,888,888,888,
166 B 888,888,888,888,888,888,888,888,888,888),T
167 209 CALL CC9(I)
168 GO TO 41
169 210 CALL CC10(I)
170 GO TO 41
171 211 CALL CC11(I)
172 GO TO 41
173 212 CALL CC12(I)
174 GO TO 41
175 213 CALL CC13(I)
176 GO TO 41
177 214 CALL CC14(I)
178 GO TO 41
179 215 CALL CC15(I)
180 GO TO 41
181 216 CALL CC16(I)
182 GO TO 41
183 217 CALL CC17(I)
184 GO TO 41
185 218 CALL CC18(I)
186 GO TO 41
187 219 CALL CC19(I)
188 GO TO 41
189 221 CALL CC21(I)
190 GO TO 41
191 222 CALL CC22(I)
192 GO TO 41
193 223 CALL CC23(I)
194 GO TO 41
195 224 CALL CC24(I)
196 GO TO 41
197 225 CALL CC25(I)
198 GO TO 41
199 227 CALL CC27(I)
200 GO TO 41
201 229 CALL CC29(I)
202 GO TO 41
203 231 CALL CC31(I)
204 GO TO 41
205 244 CALL CC44(I)
206 GO TO 41
207 245 CALL CC45(I)
208 GO TO 41
209 248 CALL CC48(I)
210 GO TO 41
211 249 CALL CC49(I)
212 GO TO 41
213 258 CALL CC58(I)
214 GO TO 41
215 259 CALL CC59(I)
216 GO TO 41
217 261 CALL CC61(I)
218 GO TO 41
219 263 CALL CC63(I)
220 GO TO 41
221 264 CALL CC64(I)
222 GO TO 41

```



```

223      265 CALL CC65(I)
224      GO TO 41
225      272 CALL CC72(I)
226      GO TO 41
227      274 CALL CC74(I)
228      GO TO 41
229      276 CALL CC76(I)
230      GO TO 41
231      277 CALL CC77(I)
232      GO TO 41
233      279 CALL CC79(I)
234      GO TO 41
235      281 CALL CC81(I)
236      GO TO 41
237      282 CALL CC82(I)
238      GO TO 41
239      284 CALL CC84(I)
240      GO TO 41
241      286 CALL CC86(I)
242      GO TO 41
243      287 CALL CC87(I)
244      GO TO 41
245      290 CALL CC90(I)
246      GO TO 41
247      291 CALL CC91(I)
248      GO TO 41
249      297 CALL CC97(I)
250      GO TO 41
251      888 WRITE (6,1888) T
252      1888 FORMAT (1H0,10X,'ERROR SITUATION. ATTEMPT TO CALL SUBROUTINE CC',
253      1 I3,'WHERE IT IS NOT SUPPOSED TO BE SO'/11X,'NO SUCH ',
254      2 'SUBROUTINE EXISTS')
255      WRITE (7,1889) T
256      1889 FORMAT(10X,'ERROR SITUATION. ATTEMPT TO CALL SUBROUTINE CC',
257      * I3,'WHERE IT IS NOT SUPPOSED TO BE SO'/11X,'NO SUCH ',
258      * 'SUBROUTINE EXISTS')
259      GO TO 76
260      C CHECK IF THE CONDITION HAS GOT ITS VALUE NOW
261      CD 41 WRITE (6,444) KGLOB,DATA(KGLOB),PRD(KGLOB)
262      CD444 FORMAT(1X,'DATA CHECK JUST BEFORE STATEMENT 41 IS SPECK'//
263      CD 120X,I10,1X,F14.4,5X,L7)
264      41 IF (PRD(KGLOB)) GO TO 50
265      GO TO 12
266      C
267      CC FOLLOWING IS A CHECK WHETHER THE MISSING INGREDIENT
268      CC IS OBTAINABLE BY EXECUTING ANY TABLE
269      C
270      39 IF (TABD(IDATA) .NE. 0) GO TO 46
271      C
272      CC WRITE THE ERROR MESSAGE THAT THIS DATA IS NOT AVAILABLE
273      C
274      WRITE (7,112) IDATA
275      112 FORMAT (1H0,10X,'ERROR MESSAGE; DATA NUMBER',I5/
276      1'IS NOT AVAILABLE.THIS IS AN INGREDIENT OF A CONDITION')
277      NUMBER=2
278      2008 KG=IDATA
279      CALL READIN(KG,NUMBER,IDATA)
280      IF (KG.NE.IDATA) GO TO 2010
281      GO TO 2007
282      2010 WRITE(7,1020)
283      1020 FORMAT(1X,'YOU HAVE INPUT THE INCORRECT VALUE OF KG',
284      * ' PLEASE TRY AGAIN.GOOD LUCK')
285      GO TO 2008
286      46 TABNO = TABD(IDATA)
287      C
288      CC START THE STACKING PROCEDURE TO EXECUTE THE APPROPRIATE TABLE
289      CC THE VALUE OF IFLAG = 1 INDICATES THAT THE STACKING IS REQUIRED
290      CC BECAUSE OF SOME MISSING INGREDIENT OF A CONDITION
291      C
292      IFLAG = 1
293      CALL STAK(STACK,ISTACK,IFLAG,T,I,J,IR,TABNO,IDATA,TRACE)
294      GO TO 18
295      C
296      CC START THE STACKING PROCEDURE TO EXECUTE THE APPROPRIATE TABLE

```



```

297 CC THE VALUE OF IFLAG = 2 INDICATES THAT THE STACKING
298 CC IS REQUIRED BECAUSE THE MISSING CONDITION IS OBTAINABLE
299 CC BY EXECUTING SOME OTHER TABLE
300 C
301 45 TABNO = TABD(KGLOB)
302 IFLAG = 2
303 IDATA = 0
304 CALL STAK(STACK,ISTACK,IFLAG,T,I,J,KGLOB,TABNO,DATA,TRACE)
305 GO TO 18
306 C
307 CC MATCH THE RULE
308 C
309 50 IF (LARRY3(IJ) .EQ. 1 .AND. DATA(KGLOB) .NE. TRUE) GO TO 55
310 IF (LARRY3(IJ) .EQ. 2 .AND. DATA(KGLOB) .EQ. TRUE) GO TO 55
311 C
312 CC CONTINUE MATCHING CONDITIONS IN THIS RULE
313 C
314 CD 52 WRITE(6,440) I
315 CD440 FORMAT(1X,'AT STATEMENT 52 IN SPECHK,THE CONDITION NO. IS',
316 CD *5I,I3)
317 52 IF (I .EQ. N(T)) GO TO 31
318 I = I+1
319 GO TO 24
320 C CONTINUE THE SEARCH WITH THE NEXT RULE
321 55 IF (J.EQ. L(T)) GO TO 30
322 J = J+1
323 CD WRITE(6,445) J
324 CD445 FORMAT(1X,'AFTER STATEMENT 55 IN SPECHK,THE RULE NO. IS',5I,I3)
325 GO TO 19
326 C
327 CC MESSAGE FOR UNSUCCESSFUL MATCH IN THE TABLE
328 C
329 30 WRITE (6,130) T
330 130 FORMAT (1H0,10X,'NO RULE IN TABLE',I4,' IS MATCHING ',
331 1 'THE CONDITION STUB'/11X,' CYCLE TERMINATED')
332 WRITE (7,132) T
333 132 FORMAT(10X,'NO RULE IN TABLE',I4,'IS MATCHING ',
334 1 'THE CONDITION STUB'/11X,' CYCLE TERMINATED')
335 GO TO 76
336 C
337 CC THE APPLICABLE RULE HAS BEEN IDENTIFIED
338 CC CHECK IF THIS INFORMATION IS DESIRED TO BE PRINTED OUT
339 C
340 31 IF (TRACE .NE. YES) GO TO 57
341 WRITE (6,177) T,J
342 177 FORMAT (1H0,15X,'SCANNING OF TABLE ',I3,' IS COMPLETE. RULE ',
343 1 'NO.',I3,' APPLIES')
344 C
345 CC NOW FIND WHICH ACTION ENTRY IS APPLICABLE
346 CC CODE: 0 FOR NO ACTION, 1 FOR CONDITIONAL EVALUATION,
347 CC 2 FOR DIRECT EXECUTION ACTIONS AND 3 FOR THE ELSE RULE
348 C
349 57 K = 1
350 58 KJ = IBASE(T,4) + (J-1)*N(T) + K
351 IFLAG = LARRY4(KJ) + 2
352 K1 = IBASE(T,2) + K
353 GO TO (49,59,69,79,89),IFLAG
354 C
355 CC THE FOLLOWING IS A DUMMY STATEMENT AND SHOULD NEVER BE REACHED
356 C
357 49 STOP
358 C
359 CC CONDITIONAL EVALUATION
360 C
361 69 KGLOB = LARRY2(K1)
362 C
363 CC BEFORE CALLING SUBROUTINE AA OF THIS TABLE, CHECK IF
364 CC THERE ARE ANY INGREDIENTS FOR THIS ACTION.
365 CC IF YES, THEN CHECK THEIR PRESENCE
366 C
367 IF ((IPNTRA(K1+1) - IPNTRA(K1)) .EQ. 0) GO TO 93
368 IR = IPNTRA(K1) + 1
369 27 IDATA = LARRY6(IR)
370 2011 IF (.NOT. PRD(IDATA)) GO TO 35
371 16 IR = IR + 1
372 IF (IR .LE. IPNTRA(K1+1)) GO TO 27

```





```

373      E
374      EC      A NORMAL EXIT FROM THIS LOOP INDICATES THAT ALL THE
375      EC      INGREDIENTS NEEDED FOR EVALUATING THIS ACTION ARE PRESENT
376      EC      AND SO SUBROUTINE AA FOR THIS TABLE CAN BE CALLED
377      C
378      93 GO TO (999,999,503,504,505,999,507,508,509,510,
379      1      511,512,513,514,515,516,999,518,519,999,
380      2      521,522,523,524,525,526,527,528,529,530,
381      3      531,532,999,999,999,999,999,999,999,999,
382      4      999,542,543,544,545,546,547,548,549,550,
383      5      551,552,553,554,555,999,557,558,559,560,
384      6      561,562,563,564,565,566,567,568,569,570,
385      7      571,572,573,574,575,576,577,578,579,580,
386      8      581,582,999,584,585,586,587,588,589,590,
387      9      591,592,593,594,595,596,597,598,999,999,
388      A      999,999,999,999,999,999,999,999,999,999,
389      B      999,999,999,999,999,999,999,999,999,999) ,T
390
391      503 CALL AA3 (K)
392      GO TO 60
393      504 CALL AA4 (K)
394      GO TO 60
395      505 CALL AA5 (K)
396      GO TO 60
397      507 CALL AA7 (K)
398      GO TO 60
399      508 CALL AA8 (K)
400      GO TO 60
401      509 CALL AA9 (K)
402      GO TO 60
403      510 CALL AA10 (K)
404      GO TO 60
405      511 CALL AA11 (K)
406      GO TO 60
407      512 CALL AA12 (K)
408      GO TO 60
409      513 CALL AA13 (K)
410      GO TO 60
411      514 CALL AA14 (K)
412      GO TO 60
413      515 CALL AA15 (K)
414      GO TO 60
415      516 CALL AA16 (K)
416      GO TO 60
417      518 CALL AA18 (K)
418      GO TO 60
419      519 CALL AA19 (K)
420      GO TO 60
421      521 CALL AA21 (K)
422      GO TO 60
423      522 CALL AA22 (K)
424      GO TO 60
425      523 CALL AA23 (K)
426      GO TO 60
427      524 CALL AA24 (K)
428      GO TO 60
429      525 CALL AA25 (K)
430      GO TO 60
431      526 CALL AA26 (K)
432      GO TO 60
433      527 CALL AA27 (K)
434      GO TO 60
435      528 CALL AA28 (K)
436      GO TO 60
437      529 CALL AA29 (K)
438      GO TO 60
439      530 CALL AA30 (K)
440      GO TO 60
441      531 CALL AA31 (K)
442      GO TO 60
443      532 CALL AA32 (K)
444      GO TO 60
445      542 CALL AA42 (K)
446      GO TO 60
447      543 CALL AA43 (K)

```



447	GO TO 60	497	GO TO 60
448	544 CALL AA44 (K)	498	570 CALL AA70 (K)
449	GO TO 60	499	GO TO 60
450	545 CALL AA45 (K)	500	571 CALL AA71 (K)
451	GO TO 60	501	GO TO 60
452	546 CALL AA46 (K)	502	572 CALL AA72 (K)
453	GO TO 60	503	GO TO 60
454	547 CALL AA47 (K)	504	573 CALL AA73 (K)
455	GO TO 60	505	GO TO 60
456	548 CALL AA48 (K)	506	574 CALL AA74 (K)
457	GO TO 60	507	GO TO 60
458	549 CALL AA49 (K)	508	575 CALL AA75 (K)
459	GO TO 60	509	GO TO 60
460	550 CALL AA50 (K)	510	576 CALL AA76 (K)
461	GO TO 60	511	GO TO 60
462	551 CALL AA51 (K)	512	577 CALL AA77 (K)
463	GO TO 60	513	GO TO 60
464	552 CALL AA52 (K)	514	578 CALL AA78 (K)
465	GO TO 60	515	GO TO 60
466	553 CALL AA53 (K)	516	579 CALL AA79 (K)
467	GO TO 60	517	GO TO 60
468	554 CALL AA54 (K)	518	580 CALL AA80 (K)
469	GO TO 60	519	GO TO 60
470	555 CALL AA55 (K)	520	581 CALL AA81 (K)
471	GO TO 60	521	GO TO 60
472	557 CALL AA57 (K)	522	582 CALL AA82 (K)
473	GO TO 60	523	GO TO 60
474	558 CALL AA58 (K)	524	584 CALL AA84 (K)
475	GO TO 60	525	GO TO 60
476	559 CALL AA59 (K)	526	585 CALL AA85 (K)
477	GO TO 60	527	GO TO 60
478	560 CALL AA60 (K)	528	586 CALL AA86 (K)
479	GO TO 60	529	GO TO 60
480	561 CALL AA61 (K)	530	587 CALL AA87 (K)
481	GO TO 60	531	GO TO 60
482	562 CALL AA62 (K)	532	588 CALL AA88 (K)
483	GO TO 60	533	GO TO 60
484	563 CALL AA63 (K)	534	589 CALL AA89 (K)
485	GO TO 60	535	GO TO 60
486	564 CALL AA64 (K)	536	590 CALL AA90 (K)
487	GO TO 60	537	GO TO 60
488	565 CALL AA65 (K)	538	591 CALL AA91 (K)
489	GO TO 60	539	GO TO 60
490	566 CALL AA66 (K)	540	592 CALL AA92 (K)
491	GO TO 60	541	GO TO 60
492	567 CALL AA67 (K)	542	593 CALL AA93 (K)
493	GO TO 60	543	GO TO 60
494	568 CALL AA68 (K)	544	594 CALL AA94 (K)
495	GO TO 60	545	GO TO 60
496	569 CALL AA69 (K)	546	595 CALL AA95 (K)



```

547      GO TO 60
548      596 CALL AA96(K)
549      GO TO 60
550      597 CALL AA97(K)
551      GO TO 60
552      598 CALL AA98(K)
553      GO TO 60
554      999 WRITE (6,1999) T
555      1999 FORMAT (1H0,10X,'ERROR SITUATION. ATTEMPT TO CALL SUBROUTINE AA',
556      1 I3,'WHERE IT IS NOT SUPPOSED TO BE SO'/11X,'NO SUCH ',
557      2 'SUBROUTINE EXISTS')
558      WRITE (7,3000) T
559      3000 FORMAT(10X,'ERROR SITUATION. ATTEMPT TO CALL SUBROUTINE AA',
560      * I3,'WHERE IT IS NOT SUPPOSED TO BE SO'/11X,'NO SUCH ',
561      * ' SUBROUTINE EXISTS')
562      GO TO 76
563      C      CHECK IF THIS ACTION IS COMPLETE
564      60 IF (KGLOBAL .EQ. 0) GO TO 59
565      IF (PRD(KGLOBAL)) GO TO 56
566      C
567      CC      ERROR MESSAGE
568      C
569      WRITE (6,141) K,T,J
570      141 FORMAT (1H0,10X,'ACTION NUMBER',I3,' OF TABLE NUMBER',I3,
571      1 ' CAN NOT BE COMPLETED.'/5X,' THE CURRENT RULE NUMBER IS',I3,
572      2 ' FURTHER EXECUTION WILL HAVE TO STOP')
573      WRITE (7,143) K,T,J
574      143 FORMAT(10X,'ACTION NUMBER',I3,' OF TABLE NUMBER',I3,
575      * ' CAN NOT BE COMPLETED.'/5X,' THE CURRENT RULE NUMBER IS',I3,
576      * ' FURTHER EXECUTION WILL HAVE TO STOP')
577      GO TO 76
578      C
579      CC      THE FOLLOWING APPLIES IF THE MISSING INGREDIENT OF THE ACTION
580      CC      IS ADDRESSED TO SOME TABLE FROM WHICH IT CAN BE RETRIEVED
581      C
582      35 IF (TABD(IDATA) .NE. 0) GO TO 36
583      C
584      CC      ERROR MESSAGE
585      C
586      WRITE (7,181) K,T,IDATA
587      181 FORMAT (1H0,15X,'ACTION NUMBER',I3,' OF TABLE ',I3,
588      1 ' CANNOT BE COMPLETED BECAUSE DATA NUMBER',I3,' IS NOT ',
589      2 'PRESENT.'/16X,'SUBROUTINE READIN IS CALLED')
590      NUMBER=3
591      2012 KG=IDATA
592      CALL READIN(KG,NUMBER,IDATA)
593      IF (KG.NE.IDATA) GO TO 2014
594      GO TO 2011
595      2014 WRITE(7,1040)
596      1040 FORMAT(1X,'YOU HAVE INPUT THE INCORRECT VALUE OF KG',
597      * ' PLEASE TRY AGAIN.GOOD LUCK')
598      GO TO 2012
599      C
600      CC      OBTAIN THE MISSING INGREDIENT BY EXECUTING THE TABLE TABD(IR)
601      C
602      36 TABNO = TABD(IDATA)
603      C
604      CC      STACK-UP BEFORE STARTING EXECUTION OF ANOTHER TABLE
605      C
606      CALL STAK(STACK,ISTACK,IFLAG,T,K,J,IR,TABNO,IDATA,TRACE)
607      GO TO 18
608      C
609      CC      THE FOLLOWING APPLIES IN CASE OF DIRECT EXECUTION COMMANDS
610      CC      THE ADDRESS OF THE TABLE WHICH IS DESIRED TO BE EXECUTED
611      CC      IS AVAILABLE AT LARRY2(K1)
612      CC      AT LARRY2(K1)
613      C
614      79 TABNO = LARRY2(K1)
615      IR = 0
616      IDATA = 0
617      CALL STAK(STACK,ISTACK,IFLAG,T,K,J,IR,TABNO,IDATA,TRACE)
618      GO TO 18
619      C

```



```

620      CC      THE FOLLOWING APPLIES IN CASE THE ELSE RULE IS APPLICABLE
621      C
622      89 WRITE (6,189) T
623      189 FORMAT (1H0,15X,'ELSE' RULE IS APPLICABLE IN TABLE NO.',I4,
624      1 ' '. FURTHER EXECUTION IS NOT POSSIBLE')
625      WRITE (7,190) T
626      190 FORMAT (15X,'ELSE RULE IS APPLICABLE IN TABLE NO.',I4,
627      * ' FURTHER EXECUTION IS NOT POSSIBLE')
628      GO TO 76
629      C      CALL SUBROUTINE SETS TO CHECK IF K Glob BELONGS
630      C      TO A MUTUALLY EXCLUSIVE SET. IF IT DOES, SET THE
631      C      OTHER ELEMENTS OF THE SET TO NO.
632      56 CALL SETS(KGlob)
633      59 K = K + 1
634      IF (K .GT. N(T)) GO TO 61
635      GO TO 58
636
637      C
638      CC      UNSTACKING GOES AS FOLLOWS
639      C
640      61 IF (ISTACK .EQ. 0) GO TO 76
641      IFLAG = STACK(ISTACK,1)
642      T = STACK(ISTACK,2)
643      J = STACK(ISTACK,4)
644
645      C
646      CC      IF THE VALUE OF IFLAG IS 1 OR 2, THEN THE STACKING WAS DONE
647      CC      IN THE CONDITION SECTION; OTHERWISE IN THE ACTION SECTION
648      C
649      GO TO (63,64,65,66),IFLAG
650      63 I = STACK(ISTACK,3)
651      IR = STACK(ISTACK,5)
652      GO TO 176
653      64 I = STACK(ISTACK,3)
654      KGlob = STACK(ISTACK,5)
655      C      CHECK IF A TRACE OF THE UNSTACKING IS DESIRED OR NOT
656      176 IF (TRACE .NE. YES) GO TO 67
657      WRITE(6,178) T,I,J
658      178 FORMAT(1H0,15X,'RESTART EXECUTION OF TABLE ',I3,
659      *3X,'AT CONDITION',I3,3X,'OF RULE',I3)
660      GO TO 67
661      65 IR = STACK(ISTACK,5)
662      K = STACK(ISTACK,3)
663      C      CHECK IF A TRACE OF THE UNSTACKING IS DESIRED OR NOT
664      IF (TRACE .NE. YES) GO TO 67
665      WRITE(6,179) T,K,J
666      179 FORMAT(1H0,15X,'RESTART EXECUTION OF TABLE ',I3,
667      *3X,'AT ACTION',I3,3X,'OF RULE',I3)
668      67 ISTACK = ISTACK - 1
669
670      C
671      CC      FOLLOWING IS A CHECK WHETHER THE VALUE OF THE MISSING
672      CC      INGREDIENT HAS BEEN OBTAINED OR NOT
673      C
674      68 GO TO (70,71,72,59),IFLAG
675      70 I1 = IBASE(T,1) + I
676      KGlob = LARRY1(I1)
677      IDATA = LARRY5(IR)
678      IJ = IBASE(T,3) + (J-1)*N(T) + I
679      2020 IF (PRD(IDATA)) GO TO 15
680
681      C
682      CC      ERROR MESSAGE
683      C
684      WRITE (7,116) IDATA,TABD(IDATA),I,T
685      116 FORMAT (1H0,10X,'VALUE OF DATA NUMBER',I4,' COULD NOT BE OBTAINED
686      1 EVEN BY EXECUTING TABLE NUMBER',I3,'/11X,'THIS DATA IS AN ',
687      2 'INGREDIENT OF CONDITION NUMBER',I3,' OF TABLE NUMBER',I3/
688      3 11X,'SUBROUTINE READIN IS CALLED')
689      NUMBER=2
690      2016 KG=IDATA
691      CALL READIN(KG,NUMBER,IDATA)
692      IF (KG.NE.IDATA) GO TO 2018
693      GO TO 2020
694      2018 WRITE(7,1060)
695      1060 FORMAT (1X,'YOU HAVE INPUT THE INCORRECT VALUE OF KG',
696      * ' PLEASE TRY AGAIN.GOOD LUCK')
697      GO TO 2016
698      71 IJ = IBASE(T,3) + (J-1)*N(T) + I
699      2026 IF (PRD(KGlob)) GO TO 50
700      C

```





```

697 CC      ERROR MESSAGE
698 C
699      WRITE (7,117) KGLOBAL,TABD(KGLOBAL),I,T
700      117 FORMAT (1H0,10X,'DATA NUMBER',I5,' COULD NOT BE ESTABLISHED EVEN
701      1 BY EXECUTING TABLE NUMBER',I3/11X,'THIS DATA IS CONDITION ',
702      2 'NUMBER',I3,' OF TABLE NUMBER',I3/11X,'SUBROUTINE READIN',
703      3 'IS CALLED')
704      NUMBER=1
705      2022 KG=KGLOBAL
706      CALL READIN(KG,NUMBER,KGLOBAL)
707      IF (KG.NE.KGLOBAL) GO TO 2024
708      GO TO 2026
709      2024 WRITE(7,1080)
710      1080 FORMAT(1X,'YOU HAVE INPUT THE INCORRECT VALUE OF KG',
711      * ' PLEASE TRY AGAIN.GOOD LUCK')
712      GO TO 2022
713      72 K1 = IBASE(T,2) + K
714      KGLOBAL = LARRY2(K1)
715      IDATA = LARRY6(K1)
716      2032 IF (PRD(IDATA)) GO TO 16
717 C
718 CC      ERROR MESSAGE
719 C
720      WRITE (7,118) IDATA,TABD(IDATA),K,T
721      118 FORMAT (1H0,10X,'VALUE OF DATA NUMBER',I4,' COULD NOT BE OBTAINED
722      1 EVEN BY EXECUTING TABLE NUMBER',I3/11X,'THIS DATA IS AN ',
723      2 'INGREDIENT OF ACTION  NUMBER',I3,' OF TABLE NUMBER',I3/
724      3 11X,'SUBROUTINE READIN IS CALLED')
725      NUMBER=3
726      2028 KG=IDATA
727      CALL READIN(KG,NUMBER,IDATA)
728      IF (KG.NE.IDATA) GO TO 2030
729      GO TO 2032
730      2030 WRITE(7,1100)
731      1100 FORMAT(1X,'YOU HAVE INPUT THE INCORRECT VALUE OF KG',
732      * ' PLEASE TRY AGAIN.GOOD LUCK')
733      GO TO 2028
734      76 CALL OUTPUT(ICYCLE)
735      ICYCLE = ICYCLE + 1
736      GO TO 1
737      END

```



```

1      SUBROUTINE SETUP
2
3      C
4      CC THIS SUBROUTINE READS THE DECISION TABLES AND THE PROPERTIES
5      CC OF DATA FOR PERMANENT STORAGE AND STORES THEM IN COMPACTED FORM
6      C
7      EC DECLARATIONS
8      C
9      IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
10     INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDK,TFIRST,THPSET
11     COMMON /NICA/DATA,PRD
12     COMMON /HNSTUP/LARRY1,LARRY2,LARRY3,LARRY4,LARRY5,LARRY6,IBASE,
13     1 IPNTRC,IPNTRA,TABD,L,N,W,T,TFIRST
14     COMMON /HNSTIN/ISET,HEXSET,NARCA,TRACE,THEHAP
15     COMMON /STUPIN/ ICLAR,IABROV
16     COMMON /STINTL/ INDEX,ENTRY,INGR,IDEFND,THPSET
17     DIMENSION
18     1 LARRY1(600),LARRY2(600),LARRY3(5000),LARRY4(5000),IBASE(120,4),
19     2 LARRY5( 600),LARRY6( 600),IPNTRC(600),IPNTRA(600),
20     3 DATA(700),PRD(700),TABD(700),ISET(700),
21     4 IABROV(700),ICLAR(2000),
22     5 L(120),N(120),W(120),STACK(20,5),RESULT(2),
23     6 INDEX(25),ENTRY(25,40),INGR(25,12),IDEFND(700,100),THPSET(100,20)
24     DIMENSION HEXSET(150),NARCA(100)
25
26     C
27     C INITIALISE THE ARRAYS
28
29     CALL INTIAL
30     DATA C/'C'/,YES/'YES'/,NO/'NO'/
31     IBASE1 = 0
32     IBASE2 = 0
33     IBASE3 = 0
34     IBASE4 = 0
35     IBASE5 = 0
36     IBASE6 = 0
37
38     C
39     CC INPUT THE TABLES FIRST FROM DATA SET NUMBER 9
40     CC FIRST READ THE TABLE NUMBER AND ITS SIZE
41     C
42     KOUNT = 1
43     1 READ (9,101) T,LT,HT,WT
44     101 FORMAT (4I10)
45
46     C
47     C A BLANK CARD SIGNIFIES END OF THE LAST TABLE
48
49     IF (T .EQ. 0) GO TO 4
50
51     C
52     CC CHECK THAT THE TABLE NUMBER HAS NOT EXCEEDED THE DIMENSION OF
53     CC ARRAYS L,N,AND W, WHICH IS 120
54     CC OTHERWISE GIVE AN ERROR MESSAGE HERE TO THIS EFFECT
55     C
56     IF (T .LE. 120) GO TO 12
57     WRITE (6,124) T
58     124 FORMAT (20X,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION OF L,N,W
59     * WHICH IS 120.' / 20X, 'CURRENT VALUE OF T IS',I4)
60     STOP
61     12 L(T) = LT
62     N(T) = HT
63     W(T) = WT
64     IF (THEHAP .NE. YES) GO TO 13
65     C
66     PRINT OUT THE INFORMATION ON TABLE NUMBER AND ITS SIZE
67     WRITE (6,151) T,LT,HT,WT
68     151 FORMAT (1H0,10X, 'TABLE NO.',I4,', LT =',I3,', HT =',I3,
69     1 ', WT =',I3//)
70     C
71     C NOW READ THE CONDITION ENTERIES
72     C
73     13 IF (KOUNT .NE. 1) GO TO 14
74     C
75     CC STORE THE DESIGNATION OF THE FIRST TABLE READ IN AS ITABLE
76     C
77     ITABLE = T
78     KOUNT = 1000
79     DO 5 I = 1,NT
80     READ (9,102) (INGR(I,J),J=1,6),FLAG,INDEX(I),(ENTRY(I,J),J=1,LT)
81     102 FORMAT (T51,5(I4,1X),I4,A1,21,I3,5X,40I1)

```



```

76 C
77 C CHECK IF THE NEXT CARD IS A CONTINUATION OF THIS CARD
78 C
79 IF (FLAG .NE. C) GO TO 17
80 READ (9,103) (INGR(I,J),J=7,12)
81 103 FORMAT (T51,5(I4,1X),I4)
82 GO TO 19
83
84 C
85 CC INCASE THE NEXT CARD IS NOT IN CONTINUATION OF THE PREVIOUS CARD
86 C
87 17 DO 18 J = 7,12
88 INGR(I,J) = 0
89 18 CONTINUE
90 19 IF (THEHAP .NE. YES) GO TO 5
91 C PRINT THE CONDITION ENTRIES SIMULTANEOUSLY
92 WRITE (6,162) (INGR(I,J),J=1,12),INDEX(I), (ENTRY(I,J),J=1,LT)
93 162 FORMAT (T61,12I5,T11,I5,5X,40I1)
94 5 CONTINUE
95 C
96 CC FILL THE CONDITION STUB IN THE LINEAR ARRAY "LARRY1"
97 CC FILL THE CONDITION ENTRIES THE LINEAR ARRAY "LARRY3" COLUMN-WISE
98 C
99 CC THE BASE ADDRESSES FOR THIS TABLE IN LARRY1 AND LARRY3 ARE
100 CC AVAILABLE AS IBASE(T,1) AND IBASE(T,3) RESPECTIVELY
101 C
102 IBASE(T,1) = IBASE1
103 IBASE(T,3) = IBASE3
104 C FIRST FILL THE CONDITION ENTRIES IN LARRY3 COLUMN-WISE
105 DO 41 J = 1,LT
106 C BASE ADDRESS FOR THE COLUMN
107 IJJ = IBASE3 + (J-1)*NT
108 DO 41 I = 1,NT
109 IJ = IJJ + I
110 C CHECK THAT IJ IS NOT MORE THAN THE DIMENSIONED VALUE OF LARRY3
111 IF (IJ .GT. 5000) GO TO 42
112 LARRY3(IJ) = ENTRY(I,J)
113 41 CONTINUE
114 C
115 CC FILL LARRY1 WITH CCNDITION STUB, LARRY5 WITH INGREDIENTS AND
116 CC GENERATE THE DEPENDENCE LIST FOR THE CONDITIONS OF THIS TABLE
117 C
118 DO 48 I = 1,NT
119 I1 = IBASE1 + I
120 C CHECK THAT I1 IS NOT MORE THAN THE DIMENSIONED VALUE OF LARRY1
121 IF (I1 .GT. 600) GO TO 43
122 LARRY1(I1) = INDEX(I)
123 C
124 CC THE INGREDIENT LIST FOR THIS ROW OF CONDITION STUB HAS ITS
125 CC BASE ADDRESS STORED AS FOLLOWS
126 C
127 IPNTRC(I1) = IBASE5
128 DO 46 J = 1,12
129 C IF THERE IS NO INGREDIENT, IT WILL BE INDICATED BY ZERO
130 IF (INGR(I,J) .EQ. 0) GO TO 47
131 IJ = IPNTRC(I1) + J
132 C CHECK THAT THIS IS NOT MORE THAN THE DIMENSION OF LARRY5
133 IF (IJ .GT. 600) GO TO 49
134 LARRY5(IJ) = INGR(I,J)
135 C
136 CC GENERATE THE DEPENDENCE ARRAY RIGHT HERE
137 CC NOTICE THAT INDEX(I) IS DEPENDENT OF INGR(I,J)
138 C
139 KGLOB = INGR(I,J)
140 C
141 CC SEARCH IF INDEX(I) ALREADY EXISTS IN IDEPND(KGLOB,*)
142 CC IF NOT THEN PLACE IT IN; OTHERWISE SKIP IT
143 C
144 DO 45 NUM = 1,100
145 IF (IDEPND(KGLOB,NUM) .EQ. 0) GO TO 44
146 IF (IDEPND(KGLOB,NUM) .EQ. INDEX(I)) GO TO 46
147 GO TO 45
148 44 IDEPND(KGLOB,NUM) = INDEX(I)
149 GO TO 46

```



```

149      45 CONTINUE
150      C
151      CC      A NORMAL EXIT FROM THIS LOOP INDICATES THAT THE 100 ELEMENTS
152      CC      OF IDEPND(KGLOB,*) ARE ALL THERE AND SO ITS DIMENSION
153      CC      SHOULD BE INCREASED
154      C
155      99 WRITE (6,119) KGLOB
156      119 FORMAT (20X,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION OF ',
157      1 'IDEPND(' ,I4, ' )'. THIS MESSAGE PRINTED BY FORMAT '119' )
158      STOP
159      46 CONTINUE
160      47 IBASE5 = IPNTRC(I4) + J - 1
161      48 CONTINUE
162      C
163      C      KEEP THE BASE ADDRESS READY FOR THE NEXT TABLE
164      C
165      IBASE1 = IBASE1 + NT
166      IBASE3 = IBASE3 + NT*LT
167      C      READ THE ACTION ENTRIES AND PRINT THEM, IF DESIRED
168      IF (THEMAP .NE. YES) GO TO 750
169      WRITE (6,153)
170      153 FORMAT (1X )
171      750 DO 6 K = 1,NT
172      READ (9,102) (INGR(K,J),J=1,6),FLAG,INDEX(K), (ENTRY(K,J),J=1,LT)
173      C      CHECK IF THE NEXT CARD IS A CONTINUATION OF THIS CARD
174      IF (FLAG .NE. C) GO TO 25
175      READ (9,103) (INGR(K,J),J=7,12)
176      GO TO 29
177      C
178      CC      INCASE THE NEXT CARD IS NOT IN CONTINUATION OF THE PREVIOUS CARD
179      C
180      25 DO 26 J = 7,12
181      INGR(K,J) = 0
182      26 CONTINUE
183      C      PRINT THE ACTION ENTRIES ETC. SIMULTANEOUSLY, IF SO DESIRED
184      29 IF (THEMAP .NE. YES) GO TO 6
185      WRITE (6,162) (INGR(K,J),J=1,12),INDEX(K), (ENTRY(K,J),J=1,LT)
186      6 CONTINUE
187      C
188      CC      FILL THE ACTION SUB IN THE LINEAR ARRAY "LARRY2"
189      CC      FILL THE ACTION ENTRIES IN THE LINEAR ARRAY "LARRY4" COLUMN-WISE
190      C
191      CC      THE BASE ADDRESSES FOR THIS TABLE IN LARRY2 AND LARRY4 ARE
192      CC      AVAILABLE AS IBASE(T,2) AND IBASE(T,4) RESPECTIVELY
193      C
194      IBASE(T,2) = IBASE2
195      IBASE(T,4) = IBASE4
196      C      FIRST FILL THE ACTION ENTRIES IN THE LARRY4 COLUMN-WISE
197      DO 51 J = 1,LT
198      C      BASE ADDRESS FOR THE COLUMN
199      IJJ = IBASE4 + (J-1)*NT
200      DO 51 K = 1,NT
201      KJ = IJJ + K
202      C      CHECK THAT KJ IS NOT MORE THAN THE DIMENSIONED VALUE OF LARRY4
203      IF (KJ .GT. 5000) GO TO 52
204      LARRY4(KJ) = ENTRY(K,J)
205      51 CONTINUE
206      C
207      CC      FILL LARRY2 WITH ACTION SUB, LARRY6 WITH INGREDIENTS AND
208      CC      GENERATE THE DEPENDENCE LIST FOR ACTIONS OF THIS TABLE
209      C
210      DO 58 K = 1,NT
211      K1 = IBASE2 + K
212      C      CHECK THAT K1 IS NOT MORE THAN THE DIMENSIONED VALUE OF LARRY2
213      IF (K1 .GT. 600) GO TO 53
214      LARRY2(K1) = INDEX(K)
215      C      THE INGREDIENT LIST FOR THIS ROW OF ACTION SUB IS AS FOLLOWS
216      IPNTRA(K1) = IBASE6
217      DO 56 J = 1,12
218      C      IF THERE IS NO INGREDIENT, IT WILL BE INDICATED BY ZERO
219      IF (INGR(K,J) .EQ. 0) GO TO 57
220      KJ = IPNTRA(K1) + J
221      C      CHECK THAT THIS IS NOT MORE THAN THE DIMENSION OF LARRY6
222      IF (KJ .GT. 600) GO TO 59
223      LARRY6(KJ) = INGR(K,J)

```





```

224 C
225 CC GENERATE THE DEPENDENCE ARRAY RIGHT HERE
226 CC NOTICE THAT INDEX(K) IS A DEPENDENT OF INGR(K,J)
227 C
228 KLOB = INGR(K,J)
229 C
230 CC SEARCH IF INDEX(K) ALREADY EXISTS IN IDEPND(KLOB,*)
231 CC IF NOT THEN PLACE IT IN; OTHERWISE SKIP IT
232 C
233 DO 55 NUN = 1,100
234 IF (IDEPND(KLOB,NUN) .EQ. 0) GO TO 54
235 IF (IDEPND(KLOB,NUN) .EQ. INDEX(K)) GO TO 56
236 GO TO 55
237 54 IDEPND(KLOB,NUN) = INDEX(K)
238 GO TO 56
239 55 CONTINUE
240 C
241 CC A NORMAL EXIT FROM THIS LOOP INDICATES THAT THE 100 ELEMENTS
242 CC OF IDEPND(KLOB,*) ARE ALL THERE AND SO ITS DIMENSION
243 CC SHOULD BE INCREASED
244 C
245 WRITE (6,120) KLOB
246 120 FORMAT (20X,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION OF ',
247 1 'IDEPND(',I4,','). THIS MESSAGE PRINTED BY FORMAT 120')
248 STOP
249 56 CONTINUE
250 57 IBASE6 = IPNTRA(K1) + J - 1
251 58 CONTINUE
252 C KEEP THE BASE ADDRESS READY FOR THE NEXT TABLE
253 IBASE2 = IBASE2 + NT
254 IBASE4 = IBASE4 + NT*LT
255 C APPEND LOGICAL DATA TO THE DEPENDENCE LIST
256 DO 69 J = 1,LT
257 DO 68 K = 1,NT
258 IF (ENTRY(K,J) .EQ. 1) GO TO 63
259 GO TO 68
260 C
261 CC CHECK IF THIS ACTION STORES ANY VALUE IN ANY LOCATION
262 C
263 63 IF (INDEX(K) .EQ. 0) GO TO 68
264 C
265 CC INDEX(K) IS DEPENDENT OF ALL THE LOGICAL CONDITIONS WHICH ARE
266 CC NOT IMMATERIAL FOR THIS RULE
267 C
268 DO 66 I = 1,NT
269 IJ = IBASE(T,3) + (J-1)*NT + I
270 IF (LARRY3(IJ) .EQ. 0) GO TO 66
271 C
272 CC OTHERWISE SEARCH IF INDEX(K) IS ALREADY IN THE LIST OF
273 CC DEPENDENTS OF THIS CONDITION
274 C
275 I1 = IBASE(T,1) + I
276 KLOB = LARRY1(I1)
277 DO 65 NUN = 1,100
278 IF (IDEPND(KLOB,NUN) .EQ. 0) GO TO 64
279 IF (IDEPND(KLOB,NUN) .EQ. INDEX(K)) GO TO 66
280 GO TO 65
281 64 IDEPND(KLOB,NUN) = INDEX(K)
282 GO TO 66
283 65 CONTINUE
284 C A NORMAL EXIT IS AN ERROR AS BEFORE
285 WRITE (6,121) KLOB
286 121 FORMAT (20X,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION OF ',
287 1 'IDEPND(',I4,','). THIS MESSAGE PRINTED BY FORMAT 121')
288 STOP
289 66 CONTINUE
290 68 CONTINUE
291 69 CONTINUE
292 C TO READ THE NEXT TABLE
293 GO TO 1
294 C
295 CC COMPACT THE ARRAY IDEPND* INTO A LINEAR ARRAY AND NAME IT ICLEAR
296 CC BECAUSE IT WILL BE USED IN CLEARING THE EFFECT OF CHANGES OF DATA.
297 CC EACH ELEMENT IN THE LIST OF DATA WILL HAVE AN ARROW POINTING INTO
298 CC "ICLEAR"; THESE ARROWS ARE STORED AS IARROW(KLOB)

```



```

299      C
300      CC      RANGES OF THE TWO DO LOOPS BELOW ARE SAME AS THE
301      CC      TWO DIMENSIONS OF IDEPND
302      C
303      3 J1 = 0
304      DO 80 KGLOBAL = 1,700
305      IARROW(KGLOBAL) = J1
306      C
307      CC      CHECK IF KGLOBAL BELONGS TO A SET
308      C
309      IF (ISET(KGLOBAL) .EQ. 0) GO TO 82
310      NSET = ISET(KGLOBAL)
311      C
312      CC      CHECK IF KGLOBAL IS THE FIRST ELEMENT OF THE SET.
313      CC      IF YES THEN IT GOES INTO ICLEAR; OTHERWISE NOT
314      C
315      I1 = MARCA(NSET) + 1
316      IF (HEXSET(I1) .EQ. KGLOBAL) GO TO 282
317      C
318      CC      MAKE THE DEPENDENTS OF KGLOBAL SAME AS THAT OF THE FIRST
319      CC      ELEMENT OF THE SET
320      C
321      JGLOBAL = HEXSET(I1)
322      DO 202 NUN = 1,100
323      IF (IDEPND(JGLOBAL,NUN) .EQ. 0) GO TO 80
324      IDEPND(KGLOBAL,NUN) = IDEPND(JGLOBAL,NUN)
325      202 CONTINUE
326      GO TO 80
327      C
328      CC      FILL THE DEPENDENTS OF THE FIRST ELEMENT OF THE SET WITH
329      CC      ALL THE POSSIBLE DEPENDENTS INDICATED FOR THE ELEMENTS
330      CC      OF THE SET
331      C
332      282 H1 = MARCA(NSET) + 2
333      H2 = MARCA(NSET + 1)
334      DO 220 ID = H1,H2
335      HH = HEXSET(ID)
336      DO 215 JN = 1,100
337      IF (IDEPND(HH,JN) .EQ. 0) GO TO 220
338      C
339      CC      SEARCH IF THIS DEPENDENT IS ALREADY IN THE LIST OF
340      CC      DEPENDENTS OF THE FIRST ELEMENT
341      C
342      DO 210 NUN = 1,100
343      IF (IDEPND(KGLOBAL,NUN) .EQ. 0) GO TO 205
344      IF (IDEPND(KGLOBAL,NUN) .EQ. IDEPND(HH,JN)) GO TO 215
345      GO TO 210
346      205 IDEPND(KGLOBAL,NUN) = IDEPND(HH,JN)
347      GO TO 215
348      210 CONTINUE
349      215 CONTINUE
350      220 CONTINUE
351      82 DO 70 J = 1,100
352      IF (IDEPND(KGLOBAL,J) .EQ. 0) GO TO 80
353      J1 = J1 + 1
354      C
355      CC      CHECK THAT J1 IS NOT MORE THAN THE DIMENSIONED VALUE OF
356      CC      THE ARRAY "ICLEAR" WHICH CURRENTLY IS 2,000
357      C
358      IF (J1 .GT. 2000) GO TO 33
359      ICLEAR(J1) = IDEPND(KGLOBAL,J)
360      GO TO 70
361      33 WRITE (6,133)
362      133 FORMAT (1H0,10X,'CAREFUL; YOU ARE EXCEEDING THE DIMENSION',
363      1 ' OF THE ARRAY "ICLEAR" WHICH IS 2000'/10X,'REMEDY IS TO',
364      2 ' INCREASE THIS DIMENSION')
365      STOP
366      70 CONTINUE
367      80 CONTINUE
368      GO TO 95
369      C
370      CC      FILL-UP THE BLANKS IN THE ARRAY IDASE
371      C
372      N DO 444 T = 1,120
373      IF (IDASE(T,1) .NE. 0 .OR. T .EQ. ITABLE) GO TO 444

```



```

374         IBASE(T,1) = IBASE1
375         IBASE(T,2) = IBASE2
376         IBASE(T,3) = IBASE3
377         IBASE(T,4) = IBASE4
378     444 CONTINUE
379 C
380 CC      FILL-UP THE BLANKS AT THE TAIL END OF THE ARRAY IPNTRC
381 C
382         I1 = I1 + 1
383         DO 445 I = I1,600
384             IPNTRC(I) = IBASE5
385     445 CONTINUE
386 C
387 CC      FILL-UP THE BLANKS AT THE TAIL END OF THE ARRAY IPNTRA
388 C
389         K1 = K1 + 1
390         DO 446 K = K1,600
391             IPNTRA(K) = IBASE6
392     446 CONTINUE
393 C
394 CC      INPUT PROPERTIES OF THE ELEMENTS OF DATA E.G. ADDRESSES
395 CC      OF TABLE NUMBERS FROM WHICH THEY CAN BE DERIVED
396 CC      AND THEIR MEMBERSHIP OF SETS ETC. FROM DATA SET NUMBER 9
397 C
398         IF (THENAP .NE. YES) GO TO 2
399 C
400 CC      HEADING FOR THE NEXT OUTPUT
401 C
402         WRITE (6,154)
403     154 FORMAT (1H1,30X, 'KGLOB',10X, 'TABDK',10X, 'NSET'//)
404         2 READ (9,105) KGLOB,TABDK,NSET
405     105 FORMAT (3I10)
406 C      A BLANK IS A SIGNAL OF END OF THIS DATA
407         IF (KGLOB .EQ. 0) GO TO 15
408         TABD(KGLOB) = TABDK
409         ISET(KGLOB) = NSET
410 C
411 CC      PRINT THIS INFORMATION ABOUT THE ELEMENTS OF DATA, IF DESIRED
412 C
413         IF (THENAP .NE. YES) GO TO 73
414         WRITE (6,155) KGLOB,TABDK,NSET
415     155 FORMAT (25X,3(I10,5X))
416 C
417 CC      GENERATE THE MUTUALLY EXCLUSIVE SETS IN A TEMPORARY ARRAY
418 CC      AND NAME IT THPSET
419 C
420         73 IF (NSET .EQ. 0) GO TO 2
421         DO 75 J = 1,20
422             IF (THPSET(NSET,J) .EQ. 0) GO TO 74
423             IF (THPSET(NSET,J) .EQ. KGLOB) GO TO 2
424             GO TO 75
425         74 THPSET(NSET,J) = KGLOB
426             GO TO 2
427         75 CONTINUE
428 C
429 CC      A NORMAL EXIT FROM THIS LOOP INDICATES THAT THE TEN ELEMENTS
430 CC      OF THPSET(NSET,*) ARE FULL AND ITS DIMENSION SHOULD BE INCREASED
431 C
432         WRITE (6,175) NSET
433     175 FORMAT (10X,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION OF THPSET(',
434         1 I2,'). THIS MESSAGE GENERATED BY FORMAT NUMBER 175')
435         STOP
436 C
437 CC      COMPACT THE CONTENTS OF THPSET IN ARRAY HEXSET.
438 CC      POINTER FROM MUTUALLY EXCLUSIVE SET TO HEXSET IS NAMED AS HARCA
439 C
440         15 J1 = 0
441         DO 78 NSET = 1,100
442             HARCA(NSET) = J1
443             DO 76 J = 1,20
444                 IF (THPSET(NSET,J) .EQ. 0) GO TO 78
445                 J1 = J1 + 1
446 C
447 CC      CHECK THAT J1 IS NOT MORE THAN THE DIMENSIONED VALUE
448 CC      OF THE ARRAY HEXSET WHICH IS CURRENTLY 150

```





```

449 C
450 IF (J1.GT. 150) GO TO 79
451 HEXSET(J1) = THPSET(HSET,J)
452 GO TO 76
453 79 WRITE (6,179)
454 179 FORMAT (1H0,10X,'CAREFUL; YOU ARE EXCEEDING THE DIMENSION ',
455 1 ' OF THE ARRAY HEXSET WHICH IS 150'/
456 211X,'REHEDY IS TO INCREASE THIS DIMENSION')
457 STOP
458 76 CONTINUE
459 78 CONTINUE
460
461 C
462 CC NOW TO COMPACT THE ARRAY IDEPND
463 C
464 GO TO 3
465 C
466 READ WHICH TABLE MUST BE EXECUTED FIRST
467 95 READ (5,107) TFIRST
468 107 FORMAT (I10)
469 GO TO 39
470 42 WRITE (6,142)
471 142 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY3',
472 1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 5000'/
473 2 11X,'REHEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
474 STOP
475 43 WRITE (6,143)
476 143 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY1',
477 1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 600'/
478 2 11X,'REHEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
479 STOP
480 49 WRITE (6,149)
481 149 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY5',
482 1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 600'/
483 2 11X,'REHEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
484 STOP
485 52 WRITE (6,152)
486 152 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY4',
487 1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 5000'/
488 2 11X,'REHEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
489 STOP
490 53 WRITE (6,150)
491 150 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY2',
492 1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 600'/
493 2 11X,'REHEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
494 STOP
495 59 WRITE (6,159)
496 159 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY6',
497 1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 600'/
498 2 11X,'REHEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
499 STOP
500 C
501 CC CHECK IF THE MAP OF THE PERMANENT STORAGE IS DESIRED
502 C
503 39 IF (THENAP.NE. YES) GO TO 999
504 WRITE (6,500) (LARRY1(I),I=1,600)
505 500 FORMAT (1H1,5X,'LARRY1'//30(5X,10I5,5X,10I5/))
506 WRITE (6,501) (LARRY3(I),I=1,5000)
507 501 FORMAT (1H1,5X,'LARRY3'//50(5X,10(10I1,1X)/))
508 WRITE (6,502) (LARRY2(I),I=1,600)
509 502 FORMAT (1H1,5X,'LARRY2'//30(5X,10I5,5X,10I5/))
510 WRITE (6,503) (LARRY4(I),I=1,5000)
511 503 FORMAT (1H1,5X,'LARRY4'//50(5X,10(10I1,1X)/))
512 WRITE (6,505) (J,(IBASE(I,J),I=1,120),J=1,4)
513 505 FORMAT (1H1/4(5X,'IBASE',I1//6(5X,10I5,5X,10I5/))//4)
514 WRITE (6,506) (LARRY5(I),I=1,600)
515 506 FORMAT (1H1,5X,'LARRY5'//30(5X,10I5,5X,10I5/))
516 WRITE (6,507) (LARRY6(I),I=1,600)
517 507 FORMAT (1H1,5X,'LARRY6'//30(5X,10I5,5X,10I5/))
518 WRITE (6,508) (IPNTRC(I),I=1,600)
519 508 FORMAT (1H1,5X,'IPNTRC'//30(5X,10I5,5X,10I5/))
520 WRITE (6,509) (IPNTRA(I),I=1,600)
521 509 FORMAT (1H1,5X,'IPNTRA'//30(5X,10I5,5X,10I5/))
522 WRITE (6,511)
523 511 FORMAT (1H1,10X,'NOW COMPACTED LIST OF DEPENDENTS'//
524 1 T10,'DATA',T17,'NO. OF',T30,'LIST OF DEPENDENTS'//
525 2 T10,'NO.',T15,'DEPENDENTS'//)

```





```

524      DO 514 I = 1,700
525      DO 513 J = 1,100
526      IF (IDEPND(I,J) .EQ. 0) GO TO 613
527      513 CONTINUE
528      613 IF (J .EQ. 1) GO TO 514
529      J1 = J-1
530      WRITE (6,512) I,J1, (IDEPND(I,J),J=1,J1)
531      512 FORMAT (10X,I3,5X,'('I2,')',4(T30,4(5I4,2X)/))
532      514 CONTINUE
533      WRITE (6,520) (IARROW(I),I=1,700)
534      520 FORMAT (1H1,5X,'IARROW'/35(5X,10I5,5X,10I5/))
535      WRITE (6,521) (ICLEAR(I),I=1,2000)
536      521 FORMAT (1H1,5X,'ICLEAR'//20(5(5X,10I5,5X,10I5/)/))
537      WRITE (6,522) ((THPSET(I,J),J=1,20),I=1,100)
538      522 FORMAT (1H1,20X,'THPSET'//5 (10(20X,10I5/)/))
539      WRITE (6,523) (HEXSET(I),I=1,150)
540      523 FORMAT (1H1,20X,'HEXSET'/15(20X,10I5/))
541      WRITE (6,524) (HARCA(I),I=1,100)
542      524 FORMAT (1H0,20X'HARCA'/5(20X,10I5/))
543      WRITE (6,525) (ISET(I),I=1,700)
544      525 FORMAT (1H1,5X,'ISET'/35(5X,10I5,5X,10I5/))
545      999 WRITE (6,172) TFIRST
546      172 FORMAT (1H0,15X,'EXECUTION WILL START WITH TABLE NO.',I4)
547      WRITE (8) LARRY1,LARRY2,LARRY5,LARRY6,IPNTRC,IPNTRA
548      WRITE (8) LARRY3
549      WRITE (8) LARRY4
550      WRITE (8) ISET,TABD,L,N,N,TFIRST,IBASE
551      WRITE (8) HEXSET,HARCA,TRACE,ICLEAB,IARROW
552      RETURN
553      END

```



```

1      SUBROUTINE INITIAL
2      CC THIS SUBROUTINE IS USED TO INITIALIZE THE VARIOUS ARRAYS
3      CC USED BY THE PROGRAM
4      IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
5      INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDK,TFIRST,THPSET
6      COMMON /NICA/DATA,PRD
7      COMMON /HNSTUP/LARRY1,LARRY2,LARRY3,LARRY4,LARRY5,LARRY6,IBASE,
8      1 IPNTRC,IPNTRA,TABD,L,H,W,T,TFIRST
9      COMMON /HNSTIN/ISET,HEXSET,HARCA,TRACE
10     COMMON /STUPIN/ ICLEAR,IARROW
11     COMMON /STINTL/ INDEX,ENTRY,INGR,IDEPND,THPSET
12     DIMENSION
13     1 LARRY1(600),LARRY2(600),LARRY3(5000),LARRY4(5000),IBASE(120,4),
14     2 LARRY5( 600),LARRY6( 600),IPNTRC(600),IPNTRA(600),
15     3 DATA(700),PRD(700),TABD(700),ISET(700),
16     4 IARROW(700),ICLEAR(2000),
17     5 L(120),H(120),W(120),STACK(20,5),RESULT(2),
18     6 INDEX(25),ENTRY(25,40),INGR(25,12),IDEPND(700,100),THPSET(100,20)
19     DIMENSION HEXSET(150),HARCA(100)
20     DO 10 I = 1,600
21     LARRY1(I) = 0
22     LARRY2(I) = 0
23     IPNTRC(I) = 0
24     IPNTRA(I) = 0
25     10 CONTINUE
26     DO 20 I = 1,700
27     DATA(I) = 0.
28     PRD(I) = .FALSE.
29     TABD(I) = 0
30     ISET(I) = 0
31     IARROW(I) = 0
32     DO 22 J = 1,100
33     IDEPND(I,J) = 0
34     22 CONTINUE
35     20 CONTINUE
36     DO 21 I = 1,1000
37     LARRY5(I) = 0
38     LARRY6(I) = 0
39     21 CONTINUE
40     DO 30 I = 1,5000
41     LARRY3(I) = 0
42     LARRY4(I) = 0
43     30 CONTINUE
44     DO 31 I = 1,2000
45     ICLEAR(I) = 0
46     31 CONTINUE
47     DO 40 I = 1,100
48     HARCA(I) = 0
49     DO 40 J = 1,20
50     THPSET(I,J) = 0
51     40 CONTINUE
52     DO 41 I = 1,120
53     L(I) = 0
54     H(I) = 0
55     W(I) = 0
56     DO 41 J = 1,4
57     IBASE(I,J) = 0
58     41 CONTINUE
59     DO 50 I = 1,25
60     INDEX(I) = 0
61     DO 48 J = 1,40
62     ENTRY(I,J) = 0
63     48 CONTINUE
64     DO 49 J = 1,12
65     INGR(I,J) = 0
66     49 CONTINUE
67     50 CONTINUE
68     DO 60 I = 1,20
69     DO 60 J = 1,5
70     STACK(I,J) = 0
71     60 CONTINUE
72     DO 70 I = 1,150
73     HEXSET(I) = 0
74     70 CONTINUE
75     RETURN
76     END

```



```

1      SUBROUTINE STAK(STACK,ISTACK,IFLAG,T,I,J,IR,TABNO,IDATA,TRACE)
2      C
3      CC THIS SUBROUTINE PERFORMS STACKING OF DECISION TABLES FOR
4      CC CONDITIONAL EXECUTION AND GENERATES MESSAGES TO THIS EFFECT
5      IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
6      INTEGER*2 STACK,ISTACK,IFLAG,T,I,J,IR,TABNO,IDATA,TABD
7      COMMON/HNSTUP/LARRY1,LARRY2,LARRY3,LARRY4,LARRY5,LARRY6,IBASE,
8      1 IPNTRC,IPNTRA,TABD,L,H,N
9      DIMENSION STACK(20,5),L(120),LARRY1(600),LARRY2(600),
10     LARRY3(5000),LARRY4(5000),LARRY5(600),LARRY6(600),
11     2IBASE(120,4),IPNTRC(600),IPNTRA(600),TABD(700),
12     3H(120),N(120)
13     DATA YES/'YES'//,NO/'NO'/
14     ISTACK = ISTACK * 1
15     STACK(ISTACK,1) = IFLAG
16     STACK(ISTACK,2) = T
17     STACK(ISTACK,3) = I
18     STACK(ISTACK,4) = H
19     STACK(ISTACK,5) = IN
20     C CHECK IF A TRACE OF THE STACKING INFORMATION IS DESIRED OR NOT
21     IF (TRACE .NE. YES) GO TO 10
22     C
23     C PRINT THE APPROPRIATE MESSAGE
24     C
25     GO TO (4,5,6,7),IFLAG
26     C
27     CC MISSING INGREDIENT OF A CONDITION
28     C
29     4 WRITE (6,104) T,I,J,IDATA,TABNO
30     104 FORMAT (1H0,10X,'SUSPENDED EXECUTION OF TABLE ',I3,
31     1 ' AT CONDITION',I3,' OF RULE',I3/11X,' REASON: MISSING ',
32     2 ' INGREDIENT CORRESPONDING TO DATA NUMBER ',I3/
33     3 11X,' STARTED EXECUTION OF TABLE ',I3)
34     GO TO 10
35     E
36     CC MISSING VALUE OF THE CONDITION ITSELF
37     E
38     5 WRITE (6,105) T,I,J,TABNO,IR
39     105 FORMAT (1H0,10X,'SUSPENDED EXECUTION OF TABLE ',I3,
40     1 ' AT CONDITION',I3,' OF RULE',I3/11X,' STARTED EXECUTION',
41     2 ' OF TABLE ',I3,' TO OBTAIN VALUE OF DATA NUMBER ',I4)
42     GO TO 10
43     E
44     CC MISSING INGREDIENT OF AN ACTION
45     C
46     6 WRITE (6,106) T,I,J,IDATA,TABNO
47     106 FORMAT (1H0,10X,'SUSPENDED EXECUTION OF TABLE ',I3,
48     1 ' AT ACTION',I3,' OF RULE',I3/11X,' REASON: MISSING ',
49     2 ' INGREDIENT CORRESPONDING TO DATA NUMBER ',I3/
50     3 11X,' STARTED EXECUTION OF TABLE ',I3)
51     GO TO 10
52     E
53     CC DIRECT EXECUTION
54     E
55     7 WRITE (6,107) T,I,J,TABNO
56     107 FORMAT (1H0,10X,'SUSPENDED EXECUTION OF TABLE ',I3,
57     1 ' AT ACTION',I3,' OF RULE',I3/11X,' STARTED EXECUTION',
58     2 ' OF TABLE ',I3,' FOR DIRECT EXECUTION')
59     10 T = TABNO
60     C IF TABLE T DOES NOT EXIST ,TERMINATE EXECUTION.
61     CD WRITE(7,110) L(T)
62     CD110 FORMAT(1X,'L(T)=' ,I4)
63     IF (L(T).EQ.0) GO TO 9999
64     TABNO = 0
65     RETURN
66     9999 WRITE (7,109) T
67     WRITE (6,109) T
68     109 FORMAT(1X,'TABLE NUMBER',2X,I4,'DOES NOT EXIST.',
69     *' EXECUTION TERMINATED.')
70     STOP
71     END

```



```

1      SUBROUTINE INPUT(ICYCLE)
2      THIS SUBROUTINE IS USED FOR READING THE DATA VALUES AND FOR
3      CLEARING THE DEPENDENT DATA FOR SECOND OR SUBSEQUENT CYCLES
4      IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
5      INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDK,TFIRST,THPSET
6      COMMON /NICA/DATA,PRD
7      COMMON/HNSTOP/LARRY1,LARRY2,LARRY3,LARRY4,LARRY5,LARRY6,IBASE,
8      1 IPNTRC,IPNTRA,TABD,L,H,N,T,TFIRST
9      COMMON /HNSTIN/ISSET,HEXSET,HARCA,TRACE,THEHAP
10     COMMON /STURIN/ ICLEAR,IARROW
11     COMMON/STINTL/INDEX,ENTRY,INGR,IDEPND,THPSET
12     COMMON/DOOP/IPDL
13     COMMON/DUMB/WARAY
14     DIMENSION
15     3 DATA(700),PRD(700),TABD(700),ISSET(700),
16     4 IARROW(700),ICLEAR(2000),
17     5 IPDL(20,2)
18     DIMENSION HEXSET(150),HARCA(100)
19
20     C      IF NOT THE FIRST CYCLE,CHECK IF THERE ARE
21     C      ANY DATA INPUT FOR THIS CYCLE
22     C
23     IF (ICYCLE.EQ.1) GO TO 9
24     WRITE (7,208)
25     208 FORMAT(1H1,10X,'PLEASE INPUT A VALUE OF 1 OR 2 FOR INDIC')
26     WRITE (7,210)
27     210 FORMAT(10X,'1 INDICATES THERE ARE FURTHER CYCLES')
28     WRITE (7,212)
29     212 FORMAT(10X,'2 INDICATES NO FURTHER CYCLES')
30     READ (4,200) INDIC
31     200 FORMAT(I4)
32
33     C      IF INDIC=1, READ IN DATA ITEMS FOR THIS CYCLE
34     C      FROM THE TERMINAL
35     C      IF INDIC=2, NO DATA ITEMS FOR THIS CYCLE,EXECUTION COMPLETED
36     IF (INDIC.EQ.1) GO TO 201
37     IF (INDIC.EQ.2) GO TO 9999
38     201 WRITE (6,170) ICYCLE
39     202 WRITE (7,203)
40     203 FORMAT(10X,'AWAITING NEXT DATA ITEM')
41     READ (4,204) KGLOBAL,DATK
42     204 FORMAT(I5,F10.0)
43
44     C      SIGNAL LAST DATA INPUT BY A ZERO INPUT
45     C
46     IF (KGLOBAL.EQ.0) GO TO 110
47     GO TO 207
48     C      READ THE NUMERICAL DATA FOR ALL THE TABLES
49     C      HEADING FOR THE NEXT PRINTED ITEM
50     C
51     5 WRITE (6,170) ICYCLE
52     170 FORMAT (1H1,15X,'THE FOLLOWING NUMERICAL DATA HAS BEEN SUPPLIED ',
53     1 'FOR CYCLE NUMBER',I3//31X,'KGLOBAL',10X,'DATK'//)
54     1 READ (5,103,END=9999) KGLOBAL,DATK
55     103 FORMAT (I5,5X,F10.0)
56     C      SIGNAL LAST CARD IN THIS DATA BY A BLANK CARD
57     IF (KGLOBAL .EQ. 0) GO TO 15
58
59     C      IF THE DATA BELONGS TO A SET, FIX THE VALUE OF ITS
60     C      ELEMENTS FIRST
61     C
62     207 IF (ISSET(KGLOBAL) .NE. 0) CALL SETS(KGLOBAL)
63     DATA(KGLOBAL) = DATK
64     PRD(KGLOBAL) = .TRUE.
65     WRITE (6,171) KGLOBAL,DATK
66     171 FORMAT (25X,I10,5X,F10.4)
67
68     C      IF THIS IS THE FIRST CYCLE THEN GO TO READ THE NEXT CARD;
69     C      OTHERWISE CLEAR THE EFFECT OF CHANGING THIS DATA FIRST
70     C
71     IF (ICYCLE .EQ. 1) GO TO 1
72     C

```





```

73 CC   FOR THE PURPOSE OF CLEARING THE EFFECTS OF READING NEW DATA, THE
74 CC   LIST OF DEPENDENTS IS STORED WITH THE FIRST ELEMENT OF THE SET
75 C
76     NSET = ISET(KGLOB)
77 CD   WRITE(6,8) KGLOB,NSET
78 CD 8  FORMAT(1X,'KGLOB=',I4,8X,'NSET=',I4)
79     IF (NSET .EQ. 0) GO TO 11
80     KGLOB = MARCA(NSET) + 1
81 CD   WRITE(6,10) KGLOB
82 CD 10 FORMAT(1X,'KGLOB=====',I4)
83 C
84 CC   FIRST FIND OUT IF THIS DATA HAS ANY DEPENDENT DATA. IF NO THEN READ
85 CC   THE NEXT DATA; IF YES THEN CLEAR IT
86 C
87     11 IF ((IARROW(KGLOB+1) - IARROW(KGLOB)) .EQ. 0) GO TO 202
88 C
89 CC   START CLEARING
90 C
91     IZ = 0
92     LG = 1
93     13 I1 = IARROW(KGLOB) + LG
94     KDEP = ICLEAR(I1)
95     IF (.NOT. PRD(KDEP)) GO TO 21
96     PRD(KDEP) = .FALSE.
97 CD   WRITE(6,2000) KDEP
98 C2000 FORMAT(1X,'KDEP=',I4)
99 C
100 CC  CHECK IF DATA ITEM KDEP HAS ANY DEPENDENT COMPONENTS. IF YES THEN
101 CC  SUSPEND CLEARING DEPENDENTS OF KGLOB BY STACKING PROCEDURE AND
102 CC  START CLEARING DEPENDENTS OF KDEP; OTHERWISE CONTINUE CLEARING
103 CC  DEPENDENTS OF KGLOB
104 C
105     IF ((IARROW(KDEP+1) - IARROW(KDEP)) .EQ. 0) GO TO 21
106     IZ = IZ + 1
107     IPDL(IZ,1) = KGLOB
108     IPDL(IZ,2) = LG
109 CD   WRITE(6,2001) KGLOB,LG
110 C2001 FORMAT(1X,'KGLOB=',I4,5X,'LG=',I3)
111     KGLOB = KDEP
112     GO TO 12
113     21 LG = LG + 1
114 CD   WRITE(6,2002) LG
115 C2002 FORMAT(1X,'LG IN LOOP 21=',I3)
116     IF (LG .LE. (IARROW(KGLOB+1) - IARROW(KGLOB))) GO TO 13
117 C
118 CC  ALL THE DEPENDENT DATA FOR THIS KGLOB HAVE BEEN CLEARED AND SO
119 CC  UNSTACKING CAN BE STARTED
120 CD   WRITE(6,2003) IZ
121 C2003 FORMAT(1X,'IZ BEFORE STATEMENT 20=',I3)
122     20 IF (IZ .EQ. 0) GO TO 202
123     KGLOB = IPDL(IZ,1)
124     LG = IPDL(IZ,2)
125     IZ = IZ - 1
126 CD   WRITE(6,2004) KGLOB,LG,IZ
127 C2004 FORMAT(1X,'VALUES AFTER STATEMENT 20,KGLOB=',I4,
128 CD   13X,'LG=',I3,3X,'IZ=',I3)
129     GO TO 21
130 C
131 CC  READ IF A TRACE OF THE TABLES EXECUTED IS DESIRED OR NOT
132 C
133     15 READ (5,105) TRACE
134     105 FORMAT(T11,A4)
135     READ IF WRITING OUT OF THE FOLLOWING ARRAYS IS DESIRED:
136 C   LARRY1,LARRY3,LARRY2,LARRY4,LARRY5,LARRY6
137 C   IBASE,IPNTRC,IPNTRA,IARROW,ICLEAR,HEXSET,
138 C   MARCA,ISET,L
139 C
140     READ(5,108) WARAY
141     108 FORMAT(T11,A4)
142     110 RETURN
143 5999 WRITE(6,2010)
144 2010 FORMAT(1X,'EXECUTION OF PROGRAM IS COMPLETED')
145     WRITE(7,2012)
146 2012 FORMAT(1X,'EXECUTION OF PROGRAM IS COMPLETED.'/
147     *1X,'COLLECT YOUR OUTPUT FROM THE COMPUTING CENTER.COME BACK SOON')
148     STOP

```



```

1      SUBROUTINE SETS(KGLOB)
2      C
3      CC THIS SUBROUTINE IS USED TO EVALUATE THE DATA IN MUTUALLY
4      CC EXCLUSIVE SETS AT THE TIME OF EXTERNAL INPUT
5      C
6      CC DECLARATIONS
7      C
8      IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
9      INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDK,TFIRST,THPSET
10     COMMON /HICA/DATA,PRD
11     COMMON /HNSTIN/ISET,HEXSET,MARCA
12     DIMENSION DATA(700),PRD(700),ISET(700)
13     DIMENSION HEXSET(150),MARCA(100)
14     C
15     CC CHECK IF KGLOB BELONGS TO A SET. IF YES THEN
16     CC FIX THE VALUES OF ALL THE ELEMENTS OF THE SET TO NO
17     C
18     NSET = ISET(KGLOB)
19     IF (NSET .EQ. 0) GO TO 99
20     I1 = MARCA(NSET) + 1
21     I2 = MARCA(NSET + 1)
22     DO 10 I = I1,I2
23     IGLOB = HEXSET(I)
24     3 DATA(IGLOB) = 0.0
25     4 PRD(IGLOB) = .TRUE.
26     10 CONTINUE
27     DATA(KGLOB)=1.0
28     99 RETURN
29     END

```



```

      READIN
1  C2345678901234567890123456789012345678901234567890123456789012
2  C
3  C      THIS SUBROUTINE IS FOR READING IN MISSING
4  C      DATA ITEMS FROM THE TERMINAL. THIS ENABLES
5  C      THE PROGRAM TO BE RUN UNDER INTERACTIVE MODE.
6  C      A LARGE NUMBER OF ABORTIVE RUNS DUE TO MISSING
7  C      DATA ITEMS CAN THEREFORE BE ELIMINATED.
8  C
9  C
10 C      SUBROUTINE READIN(KG,NUMBER,KK)
11 C      IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
12 C      COMMON/HICA/DATA,PRD
13 C      COMMON/HNSTIN/ISET,HEXSET,MARCA,TRACE,THENAP
14 C      COMMON/STUPIN/ICLEAR,IARROW
15 C      COMMON/DOOP/IPDL
16 C      COMMON/MINCE/ICYCLE
17 C      DIMENSION DATA(700),PRD(700),ISET(700),HEXSET(150),
18 C      *MARCA(100),ICLEAR(2000),IARROW(700),IPDL(20,2)
19 C
20 C      KLOSS=KK
21 C      GO TO (10,20,30),NUMBER
22 C
23 C      WRITING ON TERMINAL TO LET USER KNOW THAT
24 C      IT IS AWAITING INPUT.
25 C
26 C      10 WRITE(7,100) KG
27 C      100 FORMAT(1X,'*****AWAITING INPUT FOR DATA ITEM WITH SUBSCRIPT=',I4/
28 C      *1X,'THIS DATA ITEM IS A CONDITION.')
29 C      GO TO 500
30 C
31 C      20 WRITE(7,120) KG
32 C      120 FORMAT(1X,'*****AWAITING INPUT FOR DATA ITEM WITH SUBSCRIPT=',I4/
33 C      *1X,'THIS DATA ITEM IS A MISSING INGREDIENT OF A CONDITION')
34 C      GO TO 500
35 C
36 C      30 WRITE(7,140) KG
37 C      140 FORMAT(1X,'*****AWAITING INPUT FOR DATA ITEM WITH SUBSCRIPT=',I4/
38 C      *1X,'THIS DATA ITEM IS A MISSING INGREDIENT OF AN ACTION')
39 C      GO TO 500
40 C
41 C      READ IN MISSING DATA SUBSCRIPT AND ITS VALUE.
42 C
43 C      500 READ(4,520) KG,DATA(KG)
44 C      520 FORMAT(I10,F10.4)
45 C
46 C      IF WISH TO TERMINATE EXECUTION,INPUT A NEGATIVE VALUE OF KG
47 C
48 C      IF (KG.LT.0) GO TO 9999
49 C
50 C      CHECK IF THE CORRECT DATA SUBSCRIPT HAS BEEN INPUT
51 C
52 C      IF (KG.NE.KK) GO TO 600
53 C      SET PRESENCE OF INPUT DATA TO .TRUE.
54 C
55 C      PRD(KG)=.TRUE.
56 C      IF INPUT DATA ITEM BELONGS TO A MUTUALLY EXCLUSIVE SET,
57 C      AND IF THE DATA=.0 . SET THE OTHER DATA ITEMS
58 C      OF THAT SET TO 0.0 AND THEIR PRESENCE TO .TRUE.
59 C      IF (DATA(KG).EQ.0.0) GO TO 540
60 C      IF (ISET(KG).NE.0) CALL SETS(KG)
61 C      540 IF (ICYCLE.EQ.1) GO TO 600
62 C
63 C      WHEN A MISSING DATA ITEM IS READ IN IN
64 C      THIS ROUTINE,ANY DEPENDENTS THAT THIS DATA
65 C      ITEM HAS IS TO BE CLEARED IN CYCLES OTHER
66 C      THAN CYCLE 1.
67 C      FOR THE PURPOSE OF CLEARING THE EFFECTS OF READING NEW DATA, THE
68 C      LIST OF DEPENDENTS IS STORED WITH THE FIRST ELEMENT OF THE SET
69 C
70 C      NSET = ISET(KLOSS)
71 C      WRITE(6,620) KLOSS,NSET
72 C      620 FORMAT(1X,'KLOSS=',I4,8X,'NSET=',I4)
73 C      IF (NSET .EQ. 0) GO TO 590
74 C      KLOSS = MARCA(NSET) + 1
75 C      WRITE(6,640) KLOSS
76 C      640 FORMAT(1X,'KLOSS*****',I4)

```



```

77      C
78      CC      FIRST FIND OUT IF THIS DATA HAS ANY DEPENDENT DATA. IF NO THEN READ
79      CC      THE NEXT DATA; IF YES THEN CLEAR IT
80      C
81      590 IF ((IARROW(KLOSS+1) - IARROW(KLOSS)) .EQ. 0) GO TO 600
82      C
83      CC      START CLEARING
84      C
85      IZ = 0
86      660 LG = 1
87      680 I1 = IARROW(KLOSS) + LG
88      KDEP = ICLEAR(I1)
89      IF (.NOT. PRD(KDEP)) GO TO 740
90      PRD(KDEP) = .FALSE.
91      CD      WRITE(6,700 ) KDEP
92      CD700  FORMAT('IX,'KDEP=',I4)
93      C
94      CC      CHECK IF DATA ITEM KDEP HAS ANY DEPENDENT COMPONENTS. IF YES THEN
95      CC      SUSPEND CLEARING DEPENDENTS OF KLOSS BY STACKING PROCEDURE AND
96      CC      START CLEARING DEPENDENTS OF KDEP; OTHERWISE CONTINUE CLEARING
97      CC      DEPENDENTS OF KLOSS
98      C

99      IF ((IARROW(KDEP+1) - IARROW(KDEP)) .EQ. 0) GO TO 740
100     IZ = IZ + 1
101     IPDL(IZ,1) = KLOSS
102     IPDL(IZ,2) = LG
103     CD      WRITE(6,720) KLOSS,LG
104     CD720  FORMAT('IX,'KLOSS=',I4,5X,'LG=',I3)
105     KLOSS = KDEP
106     GO TO 660
107     740 LG = LG + 1
108     CD      WRITE(6,760) LG
109     CD760  FORMAT('IX,'LG IN LOOP 740=',I3)
110     IF (LG .LE. (IARROW(KLOSS+1) - IARROW(KLOSS))) GO TO 680
111     C
112     CC      ALL THE DEPENDENT DATA FOR THIS KLOSS HAVE BEEN CLEARED AND SO
113     CC      UNSTACKING CAN BE STARTED
114     CD      WRITE(6,780) IZ
115     CD780  FORMAT('IX,'IZ BEFORE STATEMENT 800=',I3)
116     800 IF (IZ .EQ. 0) GO TO 600
117     KLOSS = IPDL(IZ,1)
118     LG = IPDL(IZ,2)
119     IZ = IZ - 1
120     CD      WRITE(6,820) KLOSS,LG,IZ
121     CD820  FORMAT('IX,'VALUES AFTER STATEMENT 800,KLOSS=',I4,
122     CD      13X,'LG=',I3,3X,'IZ=',I3)
123     GO TO 740
124     600 RETURN
125     C
126     9999 WRITE(7,560)
127     560 FORMAT('IX,'PROGRAM EXECUTION TERMINATED BY AN INPUT OF NEGATIVE',
128     *' VALUE OF KG')
129     WRITE(6,560)
130     STOP
131     END

```





```

1      SUBROUTINE OUTPUT(ICYCLE)
2
3      C
4      CC THROUGH THIS SUBROUTINE, IT IS POSSIBLE TO OUTPUT ANY DATA
5      CC VALUE FOR CHECKING AND DIAGNOSIS
6      C
7      CC DECLARATIONS
8      C
9      IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
10     INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDK,TFIRST,THPSET
11     COMMON /MICA/DATA,PRD
12     DIMENSION DATA(700),PRD(700)
13     WRITE (6,100) ICYCLE
14     100 FORMAT (1H1,15X,'DATA VALUES AT THE END OF CYCLE NO.',I3/
15     1 16X,'ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE'
16     2//31X,'KGLOBAL',10X,'DATAK',10X,'PRD'//)
17     DO 801 KGLOBAL = 1,700
18     IF (.NOT. PRD(KGLOBAL)) GO TO 801
19     WRITE (6,800) KGLOBAL,DATA(KGLOBAL),PRD(KGLOBAL)
20     800 FORMAT (25X,I10,1X,F10.4,5X,L7)
21     801 CONTINUE
22     RETURN
23     END

```



```

1  CLEAR
2  C   THIS IS THE SUBROUTINE FOR PRINTING AND CLEARING
3  C   DATAS IF A TABLE HAS TO BE EXECUTED MORE THAN
4  C   ONCE IN A PARTICULAR RUN.
5  C   THE DATAS WHICH ARE TO BE PRINTED AND CLEARED ARE
6  C   SPECIFIED BY THE DATA STATEMENT IN THE ACTION
7  C   SUBROUTINE OF THAT TABLE.
8  C   THE NUMBER OF TIMES THAT THE TABLE IS TO BE EXECUTED
9  C   IS SPECIFIED BY NCHECK .
10 C
11 C   SUBROUTINE CLEAR(MCLEAR)
12 C   IMPLICIT LOGICAL*1(P), INTEGER*2 (I-N)
13 C   COMMON/MICA/DATA,PRD
14 C   COMMON/HNSTIN/ISSET,HEXSET,MARCA,TRACE,THEMAP
15 C   COMMON/STUPIN/ICLEAR,IARROW
16 C   COMMON/DOOP/IPDL
17 C   DIMENSION
18 C   *DATA(700),PRD(700),ISSET(700),HEXSET(150),MARCA(100),
19 C   *ICLEAR(2000),IARROW(700),IPDL(20,2),MCLEAR(15)
20 C
21 C
22 C   WRITE THE VALUE OF (DATA(501),CHECKI)
23 C
24 C   WRITE(6,100) DATA(501)
25 C   WRITE(7,100) DATA(501)
26 C   100 FORMAT(1H1,'***** CHECKI=',2X,F4.1,2X,'*****'///)
27 C
28 C   PRINT OUT THE DATAS THAT HAVE TO BE CLEARED BEFORE
29 C   REENTERING THE TABLE.
30 C
31 C   WRITE(6,102)
32 C   WRITE(7,102)
33 C   102 FORMAT(31X,'KGLOBAL',10X,'DATA(KGLOBAL)',10X,'PRD(KGLOBAL)'///)
34 C   DO 5 J=1,15
35 C   KGLOBAL=MCLEAR(J)
36 C   IF (KGLOBAL.EQ.0) GO TO 106
37 C   WRITE(6,104) KGLOBAL,DATA(KGLOBAL),PRD(KGLOBAL)
38 C   WRITE(7,104) KGLOBAL,DATA(KGLOBAL),PRD(KGLOBAL)
39 C   104 FORMAT(25X,I10,5X,F14.4,10X,L7)
40 C   5 CONTINUE
41 C
42 C
43 C   105 I=0
44 C   10 I=I+1
45 C   KGLOBAL=MCLEAR(I)
46 C   IF KGLOBAL IS 0 , ALL OF THE DATA HAVE BEEN CLEARED
47 C   IF (KGLOBAL.EQ.0) GO TO 30
48 C

```



```

49 C   SET KLOB TO ZERO AND .FALSE.
50 C
51 DATA(KLOB)=0.0
52 PRD(KLOB)=.FALSE.
53 C
54 C   FOR THE PURPOSE OF CLEARING THE DATA, THE LIST OF
55 C   DEPENDENTS IS STORED WITH THE FIRST ELEMENT OF THE SET.
56 C
57 NSET=ISET(KLOB)
58 IF(NSET.EQ.0) GO TO 11
59 KLOB=MARCA(NSET)+1
60 C
61 C   FIND OUT IF THIS DATA HAS ANY DEPENDENT DATA. IF NO ,
62 C   READ THE NEXT DATA. IF YES, THEN CLEAR IT.
63 C
64 11 IF((IARROW(KLOB+1)-IARROW(KLOB)).EQ.0) GO TO 10
65 C
66 C   START CLEARING
67 C
68 IZ=0
69 12 LG=1
70 13 I1=IARROW(KLOB)+LG
71 KDEP=ICLEAR(I1)
72 IF(.NOT.PRD(KDEP)) GO TO 21
73 PRD(KDEP)=.FALSE.
74 C
75 C   CHECK IF DATA ITEM KDEP HAS ANY DEPENDENT COMPONENTS.
76 C   IF YES THEN SUSPEND CLEARING DEPENDENTS OF KLOB BY
77 C   STACKING PROCEDURE AND START CLEARING DEPENDENTS OF KDEP;
78 C   OTHERWISE CONTINUE CLEARING DEPENDENTS OF KLOB.
79 C
80 IF((IARROW(KDEP+1)-IARROW(KDEP)).EQ.0) GO TO 21
81 IZ=IZ+1
82 IPDL(IZ,1)=KLOB
83 IPDL(IZ,2)=LG
84 KLOB=KDEP
85 GO TO 12
86 21 LG=LG+1
87 IF(LG.LE.(IARROW(KLOB+1)-IARROW(KLOB))) GO TO 13
88 C
89 C   ALL THE DEPENDENT DATA FOR THIS KLOB HAVE BEEN CLEARED,
90 C   SO UNSTACKING CAN BE STARTED.
91 C
92 IF(IZ.EQ.0) GO TO 10
93 KLOB=IPDL(IZ,1)
94 LG=IPDL(IZ,2)
95 IZ=IZ-1
96 GO TO 21
97 30 RETURN
98 END

```



## APPENDIX E

### BATCH MODE PROCESSING PROGRAM

#### - SOURCE LISTING

This appendix contains a source listing of the program for batch mode processing of design specifications.





## SLIST BATCHMODE

```

CC THIS IS THE MAIN SUBROUTINE DOING THE BULK OF DECISION TABLE
CC PROCESSING INCLUDING IDENTIFYING THE APPLICABLE RULE,
CC AND CHECKING THE PRESENCE OF DATA ETC.
CC
CC DECLARATIONS
C
C IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
C INTCGPR2 STACK,ENTRY1,TABNO,TABD,TABDK,TPIRST,THPSET
C COMMON /MICA/DATA,PRD
C COMMON /HNSUTR/LARRY1,LARRY2,LARRY3,LARRY5,LARRY6,IBASE,
C 1 IPNTRC,IPNTRA,TABD,L,N,M,I,IFIRST
C COMMON /HNSUTR/ICSET,REXSET,HANCA,TRACE,THEAP
C COMMON /STUPIM/ICLEAR,IARROW
C COMMON /STINTL/INDEX,ENTRY,INGR,IDEPRD,THPSET
C COMMON /DNRB/MRABY
C DIMENSION
C 1 LARRY1(600),LARRY2(600),LARRY3(5000),LARRY4(5000),IBASE(120,4),
C 2 LARRY5( 600),LARRY6( 600),IPNTRC(600),IPNTRA(600),
C 3 DATA(700),PRD(700),TABD(700),ISET(700),
C 4L(120),R(120),S(120),STACK(20,5),RESULTS(2)
C DIMENSION HEXSET(150),HANCA(100),IARROW(700),ICLEAR(2000)
C DATA YES/'YES'/,NO/'NO'/
C READ IF A MAP OF PERMANENT DATA STORAGE IS DESIRED OR NOT
C READ(2,110) THENAP
C 110 FORMAT(T11,A4)
C IF ISAVE=1,ENTER SUBROUTINE SETUP,CALCULATE DECISION TABLES
C PERMANENT DATA,THEN STORE IN UNIT 8
C IF ISAVE=2,SKIP SUBROUTINE SETUP,READ DECISION TABLES
C PERMANENT DATA FROM UNIT 8
C READ(2,100) ISAVE
C 100 FORMAT(I10)
C CALL SETDSR(8,11000,11000)
C GO TO (700,110),ISAVE
C 710 CALL INITIAL
C READ(8) LARRY1,LARRY2,LARRY5,LARRY6,IPNTRC,IPNTRA
C READ(8) LARRY3
C READ(8) LARRY4
C READ(8) ISET,TABD,L,N,M,IFIRST,IBASE
C READ(8) REXSET,HANCA,TRACE,ICLEAR,IARROW
C WHEN ISAVE=2
C READ FIRST FROM UNIT 5 AGAIN IN CASE IT IS DIFFERENT
C THAN THAT READ FROM UNIT 8
C READ(5,102) TPIRST
C 102 FORMAT(I10)
C IF HARRY EQUAL YES,WRITE THE ARRAYS IF STATEMENT 900 TO 928
C IF (HARRY.EQ.YES) GO TO 730
C WRITE (6,900) (LARRY1(I),I=1,600)
C 900 FORMAT (1H1.5X,'LARRY1',//30(5X,1015,5X,1015//))
C
C WRITE (6,902) (LARRY3(I),I=1,5000)
C 902 FORMAT (1H1.5X,'LARRY3',//50(5X,101011,1X//))
C WRITE (6,908) (LARRY2(I),I=1,600)
C 908 FORMAT (1H1.5X,'LARRY2',//30(5X,1015,5X,1015//))
C WRITE (6,906) (LARRY4(I),I=1,5000)
C 906 FORMAT (1H1.5X,'LARRY4',//50(5X,101011,1X//))
C WRITE (6,908) (J,(IBASE(I,J),I=1,120),J=1,4)
C 908 FORMAT (1H1.4(5X,'IBASE',11//6(5X,1015,5X,1015//)))
C WRITE(6,910) (LARRY5(I),I=1,600)
C 910 FORMAT (1H1.5X,'LARRY5',//30(5X,1015,5X,1015//))
C WRITE (6,912) (LARRY6(I),I=1,600)
C 912 FORMAT (1H1.5X,'LARRY6',//30(5X,1015,5X,1015//))
C WRITE (6,914) (IPNTRC(I),I=1,600)
C 914 FORMAT (1H1.5X,'IPNTRC',//30(5X,1015,5X,1015//))
C WRITE (6,916) (IPNTRA(I),I=1,600)
C 916 FORMAT (1H1.5X,'IPNTRA',//30(5X,1015,5X,1015//))
C WRITE (6,918) (IARROW(I),I=1,700)
C 918 FORMAT (1H1.5X,'IARROW',/35(5X,1015,5X,1015//))
C WRITE (6,920) (ICLEAR(I),I=1,2000)
C 920 FORMAT (1H1.5X,'ICLEAR',//20(5X,1015,5X,1015//))
C WRITE (6,922) (REXSET(I),I=1,150)
C 922 FORMAT (1H1.20X,'REXSET',/15(20X,1015//))
C WRITE (6,924) (HANCA(I),I=1,100)
C 924 FORMAT (1H0.20X,'HANCA',/5(20X,1015//))
C WRITE(6,926) (ISET(I),I=1,700)
C 926 FORMAT(1H1.5X,'ISET',/35(5X,1015,5X,1015//))
C WRITE(6,928) (L(1),I=1,120)
C 928 FORMAT(1H1.20X,'ARRAY L',/12(20X,1015//))
C GO TO 730
C 700 CALL SETUP
C 730 ICYCLE = 1
C 1 CALL INPUT(ICYCLE)
C 1 PRINT DATA AGAIN TO CHECK ITS VALIDITY
C WRITE (6,799)
C 799 FORMAT (1H1.15X,'DATA PRINTED AGAIN FOR CHECKING. ONLY ',
C 2 31X,'KGLOBAL,10X,'DATA',10X,'END'//)
C DO 801 KGLOBAL = 1,700
C IF (.NOT. PRD(KGLOBAL)) GO TO 801
C WRITE (6,800) KGLOBAL,DATA(KGLOBAL),PRD(KGLOBAL)
C 800 FORMAT (25X,110,1X,11X,4,5X,17)
C 801 CONTINUE
C
C GIVE A MESSAGE THAT EXECUTION OF CYCLE NUMBER (= ICYCLE) IS ABOUT
C TO COMMENCE
C
C WRITE (6,172) ICYCLE,TPIRST
C 172 FORMAT (1H1.10X,'CYCLE NUMBER',13, 5X,'*** START',
C 1 10X,'EXECUTION WITH TABLE',14, 5X,'***')
C INITIALISE BEFORE STARTING EXECUTION OF THE TABLES

```







```

248 GO TO 41
249 CALL CC68(I)
250 GO TO 41
251 CALL CC49(I)
252 GO TO 41
253 CALL CC58(I)
254 GO TO 41
255 CALL CC59(I)
256 GO TO 41
257 CALL CC61(I)
258 GO TO 41
259 CALL CC63(I)
260 GO TO 41
261 CALL CC64(I)
262 GO TO 41
263 CALL CC65(I)
264 GO TO 41
265 CALL CC72(I)
266 GO TO 41
267 CALL CC73(I)
268 GO TO 41
269 CALL CC74(I)
270 GO TO 41
271 CALL CC75(I)
272 GO TO 41
273 CALL CC76(I)
274 GO TO 41
275 CALL CC77(I)
276 GO TO 41
277 CALL CC79(I)
278 GO TO 41
279 CALL CC81(I)
280 GO TO 41
281 CALL CC82(I)
282 GO TO 41
283 CALL CC84(I)
284 GO TO 41
285 CALL CC86(I)
286 GO TO 41
287 CALL CC87(I)
288 GO TO 41
289 CALL CC90(I)
290 GO TO 41
291 CALL CC91(I)
292 GO TO 41
293 CALL CC97(I)
294 GO TO 41
295 WRITE (6,1888) T
1888 FORMAT (1H0,10X,'ERROR SITUATION. ATTEMPT TO CALL SUBROUTINE CC',
13,' WHERE IT IS NOT SUPPOSED TO BE SO',11X,'NO SUCH ',
2 'SUBROUTINE EXISTS')
GO TO 76
CHECK IF THE CONDITION HAS GOT ITS VALUE NOW
C 11 WRITE (6,444) KGLOBAL,DATA(KGLOBAL),PRD(KGLOBAL)

844 FORMAT(1X,'DATA CHECK JUST BEFORE STATEMENT 41 IN SPECKR'//
120X,I10,I,F16.4,5X,I7)
IF (PRD(KGLOBAL)) GO TO 50
GO TO 12
C
C FOLLOWING IS A CHECK WHETHER THE MISSING INGREDIENT
C IS OBTAINABLE BY EXECUTING ANY TABLE
C
39 IF (TABD(IDATA) .NE. 0) GO TO 46
C
C WRITE THE ERROR MESSAGE THAT THIS DATA IS NOT AVAILABLE
C
C
WRITE (6,112) IDATA
112 FORMAT (1H0,10X,'ERROR MESSAGE: DATA NUMBER',I5,' IS NOT AVAILABLE
1 AND IS NOT OBTAINABLE BY EXECUTING ANY TABLE EITHER')
GO TO 76
C
46 TABNO = TABD(IDATA)
C
C START THE STACKING PROCEDURE TO EXECUTE THE APPROPRIATE TABLE
C THE VALUE OF IFLAG = 1 INDICATES THAT THE STACKING IS REQUIRED
C BECAUSE OF SOME MISSING INGREDIENT OF A CONDITION
C
IFLAG = 1
CALL STAK(STACK,ISTACK,IFLAG,T,I,J,IR,TABNO,IDATA,TRACE)
GO TO 18
C
C START THE STACKING PROCEDURE TO EXECUTE THE APPROPRIATE TABLE
C THE VALUE OF IFLAG = 2 INDICATES THAT THE STACKING
C IS REQUIRED BECAUSE THE MISSING CONDITION IS OBTAINABLE
C BY EXECUTING SOME OTHER TABLE
C
85 TABNO = TABD(KGLOBAL)
IFLAG = 2
IDATA = 0
CALL STAK(STACK,ISTACK,ISTACK,IFLAG,T,I,J,KGLOBAL,TABNO,IDATA,TRACE)
GO TO 18
C
C HATCH THE RULE
C
C
50 IF (LARRY3(IJ) .EQ. 1 .AND. DATA(KGLOBAL) .NE. TRUE) GO TO 55
IF (LARRY3(IJ) .EQ. 2 .AND. DATA(KGLOBAL) .EQ. TRUE) GO TO 55
C
C CONTINUE MATCHING CONDITIONS IN THIS RULE
C
52 WRITE(6,440) I
440 FORMAT(1X,'AT STATEMENT 52 IN SPECKR,THE CONDITION NO. IS',
5X,I3)
IF (I .EQ. N(T)) GO TO 31
I = I+1
GO TO 24

```





```

C      CONTINUE THE SEARCH WITH THE NEXT RULE
CC 55 IF (J.EQ. L(T)) GO TO 30
CC J = J+1
CC WRITE(6,845) J
CC 845 FORMAT(1X,'AFTER STATEMENT 55 IN SPEECH,THE RULE NO. IS',5X,I3)
CC GO TO 19
C
CC MESSAGE FOR UNSUCCESSFUL MATCH IN THE TABLE
C
CC 30 WRITE(6,130) T
CC 130 FORMAT(1H0,10X,'NO RULE IN TABLE',14,' IS MATCHING ',
C 1,'THE CONDITION SUB',11X,' CYCLE TERMINATED')
CC GO TO 76
C
CC THE APPLICABLE RULE HAS BEEN IDENTIFIED
CC CHECK IF THIS INFORMATION IS DESIRED TO BE PRINTED OUT
C
CC 31 IF (TRACE .NE. YES) GO TO 57
CC WRITE(6,177) T,J
CC 177 FORMAT(1H0,15X,'SCANNING OF TABLE',I3,' IS COMPLETE. RULE ',
C 1,'NO.',I3,' APPLIES')
C
CC NOW FIND WHICH ACTION ENTRY IS APPLICABLE
CC CODE: 0 FOR NO ACTION, 1 FOR CONDITIONAL EVALUATION,
CC 2 FOR DIRECT EXECUTION ACTIONS AND 3 FOR THE ELSE RULE
C
CC 57 K = 1
CC 58 KJ = IBASE(T,4) + (J-1)*H(T) + K
CC IFLAG = LARRY6(KJ) + 2
CC K1 = IBASE(T,2) + K
CC GO TO (49,59,69,79,89),IFLAG
C
CC THE FOLLOWING IS A DUMMY STATEMENT AND SHOULD NEVER BE REACHED
C
CC 49 STOP
C
CC CONDITIONAL EVALUATION
C
CC 69 KGLOBAL = LARRY2(K1)
C
CC BEFORE CALLING SUBROUTINE AA OF THIS TABLE, CHECK IF
CC THERE ARE ANY INGREDIENTS FOR THIS ACTION.
CC IF YES, THEN CHECK THEIR PRESENCE
C
CC IF ((IPWTRA(K1+1) - IPWTRA(K1)) .EQ. 0) GO TO 93
CC IR = IPWTRA(K1) + 1
CC 27 IDATA = LARRY6(IR)
CC IF (-BOT. PRD(IDATA)) GO TO 35
CC 16 IR = IR + 1
CC IF (IR .LE. IPWTRA(K1+1)) GO TO 27

```

```

C      A NORMAL EXIT FROM THIS LOOP INDICATES THAT ALL THE
CC INGREDIENTS NEEDED FOR EVALUATING THIS ACTION ARE PRESENT
CC AND SO SUBROUTINE AA FOR THIS TABLE CAN BE CALLED
C
CC 93 GO TO(999,999,503,504,505,999,507,508,509,510,
C 511,512,513,514,515,516,999,518,519,999,
C 521,522,523,524,525,526,527,528,529,530,
C 531,532,999,999,999,999,999,999,999,999,
C 999,542,543,544,545,546,547,548,549,550,
C 551,552,553,554,555,999,557,558,559,560,
C 561,562,563,564,565,566,567,568,569,570,
C 571,572,573,574,575,576,577,578,579,580,
C 581,582,999,584,585,586,587,588,589,590,
C 591,592,593,594,595,596,597,598,999,999,
C 999,999,999,999,999,999,999,999,999,999,
C 999,999,999,999,999,999,999,999,999,999).2
C
CC 503 CALL AA3(K)
CC GO TO 60
CC 504 CALL AA4(K)
CC GO TO 60
CC 505 CALL AA5(K)
CC GO TO 60
CC 507 CALL AA7(K)
CC GO TO 60
CC 508 CALL AA8(K)
CC GO TO 60
CC 509 CALL AA9(K)
CC GO TO 60
CC 510 CALL AA10(K)
CC GO TO 60
CC 511 CALL AA11(K)
CC GO TO 60
CC 512 CALL AA12(K)
CC GO TO 60
CC 513 CALL AA13(K)
CC GO TO 60
CC 514 CALL AA14(K)
CC GO TO 60
CC 515 CALL AA15(K)
CC GO TO 60
CC 516 CALL AA16(K)
CC GO TO 60
CC 518 CALL AA18(K)
CC GO TO 60
CC 519 CALL AA19(K)
CC GO TO 60
CC 521 CALL AA21(K)
CC GO TO 60
CC 522 CALL AA22(K)

```





```

GO TO 60
523 CALL AR23(K)
GO TO 60
524 CALL AR24(K)
GO TO 60
525 CALL AR25(K)
GO TO 60
526 CALL AR26(K)
GO TO 60
527 CALL AR27(K)
GO TO 60
528 CALL AR28(K)
GO TO 60
529 CALL AR29(K)
GO TO 60
530 CALL AR30(K)
GO TO 60
531 CALL AR31(K)
GO TO 60
532 CALL AR32(K)
GO TO 60
533 CALL AR33(K)
GO TO 60
534 CALL AR34(K)
GO TO 60
535 CALL AR35(K)
GO TO 60
536 CALL AR36(K)
GO TO 60
537 CALL AR37(K)
GO TO 60
538 CALL AR38(K)
GO TO 60
539 CALL AR39(K)
GO TO 60
540 CALL AR40(K)
GO TO 60
541 CALL AR41(K)
GO TO 60
542 CALL AR42(K)
GO TO 60
543 CALL AR43(K)
GO TO 60
544 CALL AR44(K)
GO TO 60
545 CALL AR45(K)
GO TO 60
546 CALL AR46(K)
GO TO 60
547 CALL AR47(K)
GO TO 60
548 CALL AR48(K)
GO TO 60
549 CALL AR49(K)
GO TO 60
550 CALL AR50(K)
GO TO 60
551 CALL AR51(K)
GO TO 60
552 CALL AR52(K)
GO TO 60
553 CALL AR53(K)
GO TO 60
554 CALL AR54(K)
GO TO 60
555 CALL AR55(K)
GO TO 60
556 CALL AR56(K)
GO TO 60
557 CALL AR57(K)
GO TO 60
558 CALL AR58(K)
GO TO 60
559 CALL AR59(K)
GO TO 60
560 CALL AR60(K)
GO TO 60
561 CALL AR61(K)
GO TO 60
562 CALL AR62(K)
GO TO 60
563 CALL AR63(K)
GO TO 60
564 CALL AR64(K)
GO TO 60
565 CALL AR65(K)
GO TO 60
566 CALL AR66(K)
GO TO 60
567 CALL AR67(K)
GO TO 60
568 CALL AR68(K)
GO TO 60
569 CALL AR69(K)
GO TO 60
570 CALL AR70(K)
GO TO 60
571 CALL AR71(K)
GO TO 60
572 CALL AR72(K)
GO TO 60
573 CALL AR73(K)
GO TO 60
574 CALL AR74(K)
GO TO 60
575 CALL AR75(K)
GO TO 60
576 CALL AR76(K)
GO TO 60
577 CALL AR77(K)
GO TO 60
578 CALL AR78(K)
GO TO 60
579 CALL AR79(K)
GO TO 60
580 CALL AR80(K)
GO TO 60
581 CALL AR81(K)
GO TO 60
582 CALL AR82(K)
GO TO 60
583 CALL AR83(K)
GO TO 60
584 CALL AR84(K)
GO TO 60
585 CALL AR85(K)
GO TO 60
586 CALL AR86(K)
GO TO 60
587 CALL AR87(K)
GO TO 60
588 CALL AR88(K)
GO TO 60
589 CALL AR89(K)
GO TO 60
590 CALL AR90(K)
GO TO 60
591 CALL AR91(K)
GO TO 60
592 CALL AR92(K)
GO TO 60
593 CALL AR93(K)
GO TO 60
594 CALL AR94(K)
GO TO 60
595 CALL AR95(K)
GO TO 60
596 CALL AR96(K)
GO TO 60
597 CALL AR97(K)
GO TO 60
598 CALL AR98(K)
GO TO 60
599 WRITE (6,1999) T
1999 FORMAT (1H0,10I,'ERROR SITUATION. ATTEMPT TO CALL SUBROUTINE AA',
1 13,'WHERE IT IS NOT SUPPOSED TO BE SO'/11Z,'NO SUCH ',
2 'SUBROUTINE EXISTS')
GO TO 76
C CHECK IF THIS ACTION IS COMPLETE
60 IF (KLOB.EQ. 0) GO TO 59
IF (PRD(KLOB)) GO TO 59
C ERROR MESSAGE
CC
C WRITE (6,141) K,T,J
141 FORMAT (1H0,10I,'ACTION NUMBER',13,' OF TABLE NUMBER',13,
1 ' CAN NOT BE COMPLETED. THE CURRENT RULE NUMBER IS',13,
2 ' FURTHER EXECUTION WILL HAVE TO STOP')
GO TO 76
C
C THE FOLLOWING APPLIES IF THE MISSING INGREDIENT OF THE ACTION
IS ADDRESSED TO SOME TABLE FROM WHICH IT CAN BE RETRIEVED
CC

```



```

C
35 IF (TABD(IDATA) .NE. 0) GO TO 36
C
C ERROR MESSAGE
C
WRITE (6,181) K,T, IDATA
181 FORMAT (1H0,15X,'ACTION NUMBER',13,' OF TABLE ',13,
1 ' CANNOT BE COMPLETED BECAUSE DATA NUMBER',13,' IS NOT ',
2 'PRESENT.'/16X,'FURTHER EXECUTION OF THIS CYCLE TERMINATED')
GO TO 76
C
C OBTAIN THE MISSING INGREDIENT BY EXECUTING THE TABLE TABD(IE)
C
36 TABNO = TABD(IDATA)
C
C STACK-UP BEFORE STARTING EXECUTION OF ANOTHER TABLE
C
CALL STAR (STACK,ISTACK,IPLAG,T,K,J,IE,TABNO,IDATA,TRACE)
GO TO 10
C
C THE FOLLOWING APPLIES IN CASE OF DIRECT EXECUTION COMMANDS
C
C THE ADDRESS OF THE TABLE WHICH IS DESIRED TO BE EXECUTED
C IS AVAILABLE AT LARRY2(K1)
C AT LARRY2(K1)
C
79 TABNO = LARRY2(K1)
IR = 0
IDATA = 0
CALL STAR (STACK,ISTACK,ISTACK,IPLAG,T,K,J,IE,TABNO,IDATA,TRACE)
GO TO 18
C
C THE FOLLOWING APPLIES IN CASE THE ELSE RULE IS APPLICABLE
C
89 WRITE (6,189) T
189 FORMAT (1H0,15X,'ELSE RULE IS APPLICABLE IN TABLE NO.',14,
1 ' FURTHER EXECUTION IS NOT POSSIBLE')
GO TO 76
59 K = K + 1
IF (K .GT. M(T)) GO TO 61
GO TO 58
C
C UNSTACKING GOES AS FOLLOWS
C
61 IF (ISTACK .EQ. 0) GO TO 76
IFLAG = STACK(ISTACK,1)
T = STACK(ISTACK,2)
J = STACK (ISTACK,4)
C
C IF THE VALUE OF IFLAG IS 1 OR 2, THEN THE STACKING WAS DONE
C IN THE CONDITION SECTION; OTHERWISE IN THE ACTION SECTION
C
GO TO (63,64,65,66), IFLAG
63 I = STACK(ISTACK,3)
IR = STACK(ISTACK,5)
GO TO 176
64 I = STACK(ISTACK,3)
KLOB = STACK(ISTACK,5)
CHECK IF A TRACE OF THE UNSTACKING IS DESIRED OR NOT
176 IF (TRACE .NE. YES) GO TO 67
WRITE (6,176) T,I,J
178 FORMAT (1H0,15X,'RESTART EXECUTION OF TABLE ',13,
*3X,'AT CONDITION',13,31,'OF RULE',13)
GO TO 67
65 IR = STACK(ISTACK,5)
66 K = STACK(ISTACK,3)
CHECK IF A TRACE OF THE UNSTACKING IS DESIRED OR NOT
IF (TRACE .NE. YES) GO TO 67
WRITE (6,179) T,K,J
179 FORMAT (1H0,15X,'RESTART EXECUTION OF TABLE ',13,
*3X,'AT ACTION',13,31,'OF RULE',13)
67 ISTACK = ISTACK - 1
C
C FOLLOWING IS A CHECK WHETHER THE VALUE OF THE MISSING
C INGREDIENT HAS BEEN OBTAINED OR NOT
C
68 GO TO (70,71,72,59), IFLAG
70 I1 = IEASE(T,1) + I
KLOB = LARRY1(I1)
IDATA = LARRY5(IR)
I2 = IEASE(T,3) + (J-1) * M(T) + I
IF (PRD(IDATA)) GO TO 15
C
C ERROR MESSAGE
C
WRITE (6,116) IDATA,TABD(IDATA),I,T
116 FORMAT (1H0,10X,'VALUE OF DATA NUMBER',14,' COULD NOT BE OBTAINED
1 EVEN BY EXECUTING TABLE NUMBER',13,11X,'THIS DATA IS AN ',
2 'INGREDIENT OF CONDITION NUMBER',13,' OF TABLE NUMBER',13/
3 11X,'FURTHER EXECUTION OF THIS CYCLE IS STOPPED')
GO TO 76
71 IJ = IEASE(T,3) + (J-1) * M(T) + I
IF (PRD(KLOB)) GO TO 50
C
C ERROR MESSAGE
C
WRITE (6,117) KLOB,TABD(KLOB),I,T
117 FORMAT (1H0,10X,'DATA NUMBER',15,' COULD NOT BE ESTABLISHED EVEN
1 BY EXECUTING TABLE NUMBER',13,11X,'THIS DATA IS CONDITION ',
2 'NUMBER',13,' OF TABLE NUMBER',13,11X,'FURTHER EXECUTION OF THIS
3 CYCLE IS STOPPED')

```



















```

C      A NORMAL EXIT IS AN ERROR AS BEFORE
      WRITE (6,121) KGLOBAL
121 FORMAT (20I,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION OF ',
     1 'IDEPND(','Ia',''). THIS MESSAGE PRINTED BY FORMAT 121')
      STOP
66 CONTINUE
68 CONTINUE
69 CONTINUE
C      TO READ THE NEXT TABLE
      GO TO 1

C      COMPACT THE ARRAY IDEPND INTO A LINEAR ARRAY AND NAME IT ICLEAR
      BECAUSE IT WILL BE USED IN CLEARING THE EFFECT OF CHANGES OF DATA.
      EACH ELEMENT IN THE LIST OF DATA WILL HAVE AN ARROW POINTING INTO
      "ICLEAR"; THESE ARROWS ARE STORED AS IARROW(KGLOBAL)
C
C      RANGES OF THE TWO DO LOOPS BELOW ARE SAME AS THE
      TWO DIMENSIONS OF IDEPND
C
3 J1 = 0
DO 80 KGLOBAL = 1,700
  IARROW(KGLOBAL) = J1
C
C      CHECK IF KGLOBAL BELONGS TO A SET
C
      IF (ISET(KGLOBAL) .EQ. 0) GO TO 82
      NSET = ISET(KGLOBAL)
C
C      CHECK IF KGLOBAL IS THE FIRST ELEMENT OF THE SET.
      IF YES THEN IT GOES INTO ICLEAR; OTHERWISE NOT
C
      I1 = MARCA(NSET) + 1
      IF (NEXSET(I1) .EQ. KGLOBAL) GO TO 282
C
C      MAKE THE DEPENDENTS OF KGLOBAL SAME AS THAT OF THE FIRST
      ELEMENT OF THE SET
C
      KGLOBAL = NEXSET(I1)
      DO 102 NUB = 1,100
      IF (IDEPND(KGLOBAL,NUB) .EQ. 0) GO TO 80
      IDEPND(KGLOBAL,NUB) = IDEPND(KGLOBAL,NUB)
102 CONTINUE
      GO TO 80
C
C      FILL THE DEPENDENTS OF THE FIRST ELEMENT OF THE SET WITH
      ALL THE POSSIBLE DEPENDENTS INDICATED FOR THE ELEMENTS
      OF THE SET
C
      282 N1 = MARCA(NSET) + 2
      N2 = MARCA(NSET) + 1
C
      DO 220 ID = N1,N2
      NM = NEXSET(ID)
      DO 215 JM = 1,100
      IF (IDEPND(NM,JM) .EQ. 0) GO TO 220
C
      SEARCH IF THIS DEPENDENT IS ALREADY IN THE LIST OF
      DEPENDENTS OF THE FIRST ELEMENT
C
      DO 210 MUN = 1,100
      IF (IDEPND(KGLOBAL,MUN) .EQ. IDEPND(NM,JM)) GO TO 215
      GO TO 210
      205 IDEPND(KGLOBAL,MUN) = IDEPND(NM,JM)
      GO TO 215
      210 CONTINUE
      215 CONTINUE
      220 CONTINUE
      82 DO 70 J = 1,100
      IF (IDEPND(KGLOBAL,J) .EQ. 0) GO TO 80
      J1 = J1 + 1
C
C      CHECK THAT J1 IS NOT MORE THAN THE DIMENSIONED VALUE OF
      THE ARRAY "ICLEAR" WHICH CURRENTLY IS 2,000
C
      IF (J1 .GT. 2000) GO TO 33
      ICLEAR(J1) = IDEPND(KGLOBAL,J)
      GO TO 70
      33 WRITE (6,133)
133 FORMAT (10I,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION',
     1 ' OF THE ARRAY "ICLEAR" WHICH IS 2000.'/10I,'REMEDY IS TO',
     2 ' INCREASE THIS DIMENSION')
      STOP
      70 CONTINUE
      80 CONTINUE
      GO TO 95
C
C      FILL-UP THE BLANKS IN THE ARRAY IBASE
C
      DO 444 T = 1,120
      IF (IBASE(T,1) .NE. 0 .OR. T .EQ. ITABLE) GO TO 444
      IBASE(T,1) = IBASE1
      IBASE(T,2) = IBASE2
      IBASE(T,3) = IBASE3
      IBASE(T,4) = IBASE4
444 CONTINUE
C
C      FILL-UP THE BLANKS AT THE TAIL END OF THE ARRAY IPTRC
C
      I1 = I1 + 1
      DO 445 I = 1,600

```



```

1 12,*) - THIS MESSAGE GENERATED BY FORMAT NUMBER 175')
STOP
C
CC COMPACT THE CONTENTS OF THPSET IN ARRAY HXSET.
CC POINTER FROM MUTUALLY EXCLUSIVE SET TO HXSET IS NAMED AS MARCA
C
15 J1 = 0
DO 78 HSET = 1,100
  MARCA(HSET) = J1
DO 76 J = 1,20
  IF (THPSET(HSET,J) .EQ. 0) GO TO 78
  J1 = J1 + 1
C
CC CHECK THAT J1 IS NOT MORE THAN THE DIMENSIONED VALUE
CC OF THE ARRAY HXSET WHICH IS CURRENTLY 150
C
IF (J1 .GT. 150) GO TO 79
HXSET(J1) = THPSET(HSET,J)
GO TO 76
79 WRITE (6,179)
179 FORMAT (1H0,10X,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION ',
1' OF THE ARRAY HXSET WHICH IS 150',
21X,'REMEDY IS TO INCREASE THIS DIMENSION')
STOP
76 CONTINUE
78 CONTINUE
C
CC HOW TO COMPACT THE ARRAY IDEPND
C
GO TO 3
C
95 READ (5,107) TPIST
107 FORMAT (I10)
GO TO 39
42 WRITE (6,142)
142 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LABRY3',
1' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 5000',
21X,'REMEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
STOP
43 WRITE (6,143)
143 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LABRY1',
1' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 600',
21X,'REMEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
STOP
49 WRITE (6,149)
149 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LABRY5',
1' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 600',
21X,'REMEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
STOP
52 WRITE (6,152)

```

```

IPWTRC(I) = IBASES
845 CONTINUE
C
CC PULL-UP THE BLANKS AT THE TAIL END OF THE ARRAY IPWTRA
C
K1 = K1 + 1
DO 846 K = K1,600
  IPWTRA(K) = IBASE6
846 CONTINUE
C
CC INPUT PROPERTIES OF THE ELEMENTS OF DATA E.G. ADDRESSES
CC OF TABLE NUMBERS FROM WHICH THEY CAN BE DERIVED
CC AND THEIR MEMBERSHIP OF SETS ETC. FROM DATA SET NUMBER 9
C
IF (THEMAE .NE. YES) GO TO 2
C
CC HEADING FOR THE NEXT OUTPUT
C
WRITE (6,154)
154 FORMAT (1H1,30X,'KGLOBAL',10X,'TABDK',10X,'HSET'//)
2 READ (9,105) KGLOBAL,TABDK,HSET
105 FORMAT (J110)
C
A BLANK IS A SIGNAL OF END OF THIS DATA
IF (KGLOBAL .EQ. 0) GO TO 15
TABD(KGLOBAL) = TABDK
ISET(KGLOBAL) = HSET
C
CC PRINT THIS INFORMATION ABOUT THE ELEMENTS OF DATA, IF DESIRED
C
IF (THEMAP .NE. YES) GO TO 73
WRITE (6,155) KGLOBAL,TABDK,HSET
155 FORMAT (25X,3(110,5X))
C
CC GENERATE THE MUTUALLY EXCLUSIVE SETS IN A TEMPORARY ARRAY
CC AND NAME IT THPSET
C
73 IF (HSET .EQ. 0) GO TO 2
DO 75 J = 1,20
  IF (THPSET(HSET,J) .EQ. 0) GO TO 74
  IF (THPSET(HSET,J) .EQ. KGLOBAL) GO TO 2
GO TO 75
74 THPSET(HSET,J) = KGLOBAL
GO TO 2
75 CONTINUE
C
CC A NORMAL EXIT FROM THIS LOOP INDICATES THAT THE TEN ELEMENTS
CC OF THPSET(HSET,*) ARE FULL AND ITS DIMENSION SHOULD BE INCREASED
C
WRITE (6,175), HSET
175 FORMAT (10I,'CAREFUL: YOU ARE EXCEEDING THE DIMENSION OF THPSET(',

```





```

152 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY4',
1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 5000',
2 11X,'REMEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
STOP
53 WRITE (6,150)
150 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY2',
1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 600',
2 11X,'REMEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
STOP
59 WRITE (6,159)
159 FORMAT (1H0,10X,'CAREFUL, THE DIMENSION OF THE ARRAY LARRY6',
1 ' IS BEING EXCEEDED. CURRENT SPECIFIED DIMENSION = 600',
2 11X,'REMEDY IS TO INCREASE THIS DIMENSION. JOB TERMINATED')
STOP
C CHECK IF THE MAP OF THE PERMANENT STORAGE IS DESIRED
CC
CC
39 IF (REMAP .NE. YES) GO TO 999
WRITE (6,500) (LARRY1(I), I=1,600)
500 FORMAT (1H1,5X,'LARRY1',/30(5X,1015,5X,1015/))
501 WRITE (6,501) (LARRY3(I), I=1,5000)
WRITE (6,501) (LARRY3(I), I=1,5000)
501 FORMAT (1H1,5X,'LARRY3',/50(5X,10(1011,1X)/))
WRITE (6,502) (LARRY2(I), I=1,600)
502 FORMAT (1H1,5X,'LARRY2',/30(5X,1015,5X,1015/))
WRITE (6,503) (LARRY4(I), I=1,5000)
503 FORMAT (1H1,5X,'LARRY4',/50(5X,10(1011,1X)/))
WRITE (6,505) (J, (IBASE(I,J), I=1,120), J=1,4)
505 FORMAT (1H1/4(5X,'IBASE',I1//6(5X,1015,5X,1015/)))
WRITE(6,506) (LARRY5(I), I=1,600)
506 FORMAT (1H1,5X,'LARRY5',/30(5X,1015,5X,1015/))
WRITE (6,507) (LARRY6(I), I=1,600)
507 FORMAT (1H1,5X,'LARRY6',/30(5X,1015,5X,1015/))
WRITE (6,508) (IPTRC(I), I=1,600)
508 FORMAT (1H1,5X,'IPTRC',/30(5X,1015,5X,1015/))
WRITE (6,509) (IPNTRA(I), I=1,600)
509 FORMAT (1H1,5X,'IPNTRA',/30(5X,1015,5X,1015/))
WRITE (6,511)
511 FORMAT (1H1,10X,'NON COMPACTED LIST OF DEPENDENTS',/
1 T10,'DATA',T17,'NO. OF',T30,'LIST OF DEPENDENTS',/
2 T10,'NO.',T15,'DEPENDENTS',/
DO 514 I = 1,700
DO 513 J = 1,100
IF (IDPEND(I,J) .EQ. 0) GO TO 613
513 CONTINUE
613 IF (J .EQ. 1) GO TO 514
J1 = J-1
WRITE (6,512) I,J1,(IDPEND(I,J), J=1,J1)
512 FORMAT (10X,I3,5X,'(',I2,',',4(T30,4(514,2X)/))
514 CONTINUE
WRITE (6,520) (IARROW(I), I=1,700)
520 FORMAT (1H1,5X,'IARROW',/35(5X,1015,5X,1015/))
WRITE (6,521) (ICLEAR(I), I=1,2000)
521 FORMAT (1H1,5X,'ICLEAR',/20(5(5X,1015,5X,1015/))
WRITE (6,522) ((THPSET(I,J), J=1,20), I=1,100)
522 FORMAT (1H1,20X,'THPSET',/5 (10(20X,1015/))
WRITE (6,523) (HEXSET(I), I=1,150)
523 FORMAT (1H1,20X,'HEXSET',/15(20X,1015/))
WRITE (6,524) (MARCA(I), I=1,100)
524 FORMAT (1H0,20X,'MARCA',/5(20X,1015/))
WRITE(6,525) (ISET(I), I=1,700)
525 FORMAT (1H1,5X,'ISET',/35(5X,1015,5X,1015/))
WRITE (6,172) TPIEST
172 FORMAT (1H0,15X,'EXECUTION WILL START WITH TABLE NO.',I4)
WRITE(8) LARRY1, LARRY2, LARRY5, LARRY6, IPTRC, IPNTRA
WRITE(8) LARRY3
WRITE(8) LARRY4
WRITE(8) ISET, TABD, L, N, TPIEST, IBASE
WRITE(8) HEXSET, MARCA, TRACE, ICLEAR, IARROW
RETURN
END
SUBROUTINE INITIAL
THIS SUBROUTINE IS USED TO INITIALIZE THE VARIOUS ARRAYS
USED BY THE PROGRAM
IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
INTEGER*2 STACK, ENTRY, I, TABNO, TABD, TABDK, TPIEST, THPSET
COMMON /BICA/DATA, PRD
COMMON /HNSSTOP/LARRY1, LARRY2, LARRY3, LARRY4, LARRY5, LARRY6, IBASE,
1 IPNTRC, IPNTRA, TABD, L, N, TPIEST, FIRST
COMMON /HNSFIN/ISET, HEXSET, MARCA, TRACE
COMMON /STUPIN/ ICLEAR, IARROW
COMMON /STIWTL/ INDEI, ENTHI, INGER, IDEPND, THPSET
DIMENSION
1 LARRY1(600), LARRY2(600), LARRY3(5000), LARRY4(5000), IBASE(120,4),
2 LARRY5( 600), LARRY6( 600), IPTRC(600), IPNTRA(600),
3 DATA(700), PRD(700), TABD(700), ISET(700),
4 IARROW(700), ICLEAR(2000),
5 I(120), N(120), M(120), STACK(20,5), RESULT(2),
6 INDEX(25), ENTRY(25,40), INGER(25,12), IDEPND(700,100), THPSET(100,20)
DIMENSION HEXSET(150), MARCA(100)
DO 10 I = 1,600
LARRY1(I) = 0
LARRY2(I) = 0
IPTRC(I) = 0
IPNTRA(I) = 0
DO 20 I = 1,700
DATA(I) = 0.
PRD(I) = .FALSE.
TABD(I) = 0
ISET(I) = 0
10 CONTINUE
20 CONTINUE

```





```

INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDK,TPRST,TRPSET
COMMON/HICA/DATA,END
COMMON/HNSTUP/LARRY1,LARRY2,LARRY3,LARRY4,LARRY5,LARRY6,IDBASE,
1 IPTEC,IPNTRA,TABD,L,N,T,TPRST
COMMON/HNSTIN/ISCT,HXSET,MARCA,TRACE,THREAP
COMMON/STUJN/ICLEAR,IARROW
COMMON/STATI/INDEX,ENTRY,INGR,IDEPRD,TRPSET
COMMON/DOOP/IPDL
COMMON/DUNB/MARAY
DIMENSION
3 DATA(700),PRD(700),TABD(700),ISCT(700),
4 IARROW(700),ICLEAR(2000),
5 IPDL(20,2),
DIMENSION HXSET(150),MARCA(100)
C READ THE NUMERICAL DATA FOR ALL THE TABLES
C HEADING FOR THE NEXT PRINTED ITEM
C
9 WRITE (6,170) ICYCLE
170 FORMAT ('111,151','THE FOLLOWING NUMERICAL DATA HAS BEEN SUPPLIED ',
1 'FOR CYCLE NUMBER',I3//31X,'KGLOBAL',10X,'DATAK'//)
1 READ (5,103,END=9999) KGLOBAL,DATK
103 FORMAT (I5,X,F10.0)
C SIGNAL LAST CARD IN THIS DATA BY A BLANK CARD
IP (KGLOBAL .EQ. 0) GO TO 15
C
C IF THE DATA BELONGS TO A SET, FIX THE VALUE OF ITS
CC ELEMENTS FIRST
C
IP (ISCT(KGLOBAL) .NE. 0) CALL SETS(KGLOBAL)
DATA(KGLOBAL) = DATK
PRD(KGLOBAL) = .TRUE.
WRITE (6,171) KGLOBAL,DATK
171 FORMAT (25X,I10,X,F10.4)
C
C IF THIS IS THE FIRST CYCLE THEN GO TO READ THE NEXT CARD;
CC OTHERWISE CLEAR THE EFFECT OF CHANGING THIS DATA FIRST
C
IP (ICYCLE .EQ. 1) GO TO 1
C
C FOR THE PURPOSE OF CLEARING THE EFFECTS OF READING NEW DATA, THE
CC LIST OF DEPENDENTS IS STORED WITH THE FIRST ELEMENT OF THE SET
C
HXSET = ISCT(KGLOBAL)
WRITE (6,8) KGLOBAL,HXSET
8 FORMAT('1X,KGLOBAL=',I4,8X,'HSET=',I4)
IP (HSET .EQ. 0) GO TO 11
KGLOBAL = MARCA(HSET) + 1
WRITE (6,10) KGLOBAL
10 FORMAT('1X,KGLOBAL=====',I4)
C
CC SUBROUTINE INPUT(ICYCLE)
END
THIS SUBROUTINE IS USED FOR READING THE DATA VALUES AND FOR
CLEARING THE DEPENDENT DATA FOR SECOND OR SUBSEQUENT CYCLES
IMPLICIT LOGICAL*(F), INTEGER*2 (I-N)

```

```

CC      SUBROUTINE INPUT(CYCLE)
CC      THIS SUBROUTINE IS USED FOR READING THE DATA VALUES AND FOR
CC      CLEARING THE DEPENDENT DATA FOR SECOND OR SUBSEQUENT CYCLES
      IMPLICIT LOGICAL*1 (F), INTEGER*2 (I-N)
      END

```



```

C
CC FIRST FIND OUT IF THIS DATA HAS ANY DEPENDENT DATA. IF NO THEN READ
CC THE NEXT DATA: IF YES THEN CLEAR IT
C
11 IF ((IARROW(KGLOB+1) - IARROW(KGLOB)) .EQ. 0) GO TO 1
C
C START CLEARING
C
IZ = 0
12 LG = 1
13 I1 = IARROW(KGLOB) + LG
KDEP = ICLEAR(I1)
IF (.NOT. KDEP) GO TO 21
PRD(KDEP) = .FALSE.
WRITE(6,2000) KDEP
C2000 FORMAT('1X','KDEP=',I4)
C
C CHECK IF DATA ITEM KDEP HAS ANY DEPENDENT COMPONENTS. IF YES THEN
C SUSPEND CLEARING DEPENDENTS OF KGLOB BY STACKING PROCEDURE AND
C START CLEARING DEPENDENTS OF KDEP; OTHERWISE CONTINUE CLEARING
C DEPENDENTS OF KGLOB
C
IF ((IARROW(KDEP+1) - IARROW(KDEP)) .EQ. 0) GO TO 21
IZ = IZ + 1
IPDL(IZ,1) = KGLOB
IPDL(IZ,2) = LG
WRITE(6,2001) KGLOB,LG
C2001 FORMAT('1X','KGLOB=',I4,'5X','LG=',I3)
KGLOB = KDEP
GO TO 12
21 LG = LG + 1
C2002 WRITE(6,2002) LG
C2002 FORMAT('1X','LG IN LOOP 21=',I3)
IF (LG .LE. (IARROW(KGLOB+1) - IARROW(KGLOB))) GO TO 13
C
C ALL THE DEPENDENT DATA FOR THIS KGLOB HAVE BEEN CLEARED AND SO
C UNSTACKING CAN BE STARTED
C2003 WRITE(6,2003) IZ
C2003 FORMAT('1X','IZ BEFORE STATEMENT 20=',I3)
20 IF (IZ .EQ. 0) GO TO 1
KGLOB = IPDL(IZ,1)
LG = IPDL(IZ,2)
IZ = IZ - 1
C2004 WRITE(6,2004) KGLOB,LG,IZ
C2004 FORMAT('1X','VALUES AFTER STATEMENT 20,KGLOB=',I4,
CB '1X','LG=',I3,'3X','IZ=',I3)
GO TO 21
C
C READ IF A TRACE OF THE TABLES EXECUTED IS DESIRED OR NOT
C
15 READ (5,105) TRACE
105 FORMAT ('11,A4)
READ IF WRITING OUT OF THE FOLLOWING ARRAYS IS DESIRED:
LARRY1,LARRY3,LARRY5,LARRY7,LARRY9,LARRY11,LARRY13,LARRY15,
IBASE,IPWTEC,IPWTRA,IARROW,ICLEAR,HEXSET,
MARCA,ISET,L
C
READ(5,108) WARAY
108 FORMAT('11,A4)
RETURN
9999 WRITE(6,2010)
2010 FORMAT('1X','EXECUTION OF PROGRAM IS COMPLETED')
2010 WRITE(7,2012)
2012 FORMAT('1X','EXECUTION OF PROGRAM IS COMPLETED.'//
*11,'COLLECT YOUR OUTPUT FROM THE COMPUTING CENTER.COME BACK SOON')
STOP
END
SUBROUTINE STAR(STACK,ISTACK,IPLAG,T,I,J,IR,TABNO,IDATA,TRACE)
C
C THIS SUBROUTINE PERFORMS STACKING OF DECISION TABLES FOR
C CONDITIONAL EXECUTION AND GENERATES MESSAGES TO THIS EFFECT
C IMPLICIT LOGICAL*1 (PI), INTEGER*2 (I-M)
C INTEGERR*2 STACK,ISTACK,IPLAG,T,I,J,IR,TABNO,IDATA,TABD
C COMMON/NESTUP/LARRY1,LARRY2,LARRY3,LARRY4,LARRY5,LARRY6,IBASE,
1 IPWTEC,IPWTRA,TABD,L,M,N
C
C DIMENSION STACK(20,5),L(120),LARRY1(600),LARRY2(600),
1 LARRY3(600),LARRY4(5000),LARRY5(600),LARRY6(600),
2 IBASE(120,4),IPWTEC(600),IPWTRA(600),TABD(700),
3 H(120),M(120)
C
C DATA YES//YES',NO//NO'/
ISTACK = ISTACK + 1
STACK(ISTACK,1) = IPLAG
STACK(ISTACK,2) = T
STACK(ISTACK,3) = I
STACK(ISTACK,4) = J
STACK(ISTACK,5) = IF
CHECK IF A TRACE OF THE STACKING INFORMATION IS DESIRED OR NOT
IF (TRACE .NE. YES) GO TO 10
C
C PRINT THE APPROPRIATE MESSAGE
C
GO TO (4,5,6,7),IPLAG
C
C MISSING INGREDIENT OF A CONDITION
C
4 WRITE (6,104) T,I,J,IDATA,TABNO
104 FORMAT ('1H0,101','SUSPENDED EXECUTION OF TABLE ',I3,
1 ' AT CONDITION',I3,' OF RULE',I3/I1X,' REASON: MISSING ',
2 'INGREDIENT CORRESPONDING TO DATA NUMBER ',I3/
3 '11',' STARTED EXECUTION OF TABLE ',I3)

```



```

100 FORMAT (1H',15X','DATA VALUES AT THE END OF CYCLE NO.',I3/,
1 16X','ONLY THAT DATA WHICH HAS A VALUE IS REPRODUCED HERE',
2//31X,'KGLOBAL',10X,'DATAK',10X,'PRD'//)
DO 801 KGLOBAL = 1,700
IF (.NOT. PRD(KGLOBAL)) GO TO 801
WRITE (6,800) KGLOBAL,DATA(KGLOBAL),PRD(KGLOBAL)
800 FORMAT (25X,I10,1V,F10.4,5X,I7)
801 CONTINUE
RETURN
END
SUBROUTINE SETS(KGLOBAL)
C
C THIS SUBROUTINE IS USED TO EVALUATE THE DATA IN MUTUALLY
C EXCLUSIVE SETS AT THE TIME OF EXTERNAL INPUT
C
C DECLARATIONS
C
C IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
C INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDR,TFIRST,THESET
C COMMON /HICA/DATA,PRD
C COMMON /HINSTIN/ISSET,HEXSET,MARCA
C DIMENSION DATA(700),PRD(700),ISSET(700)
C DIMENSION HEXSET(150),MARCA(100)
C
C CHECK IF KGLOBAL BELONGS TO A SET. IF YES THEN
C FIX THE VALUES OF ALL THE ELEMENTS OF THE SET TO NO
C
HESET = ISSET(KGLOBAL)
IF (HESET.EQ.0) GO TO 99
I1 = MARCA(HESET) + 1
I2 = MARCA(HESET + 1)
DO 10 I = I1,I2
IGLOB = HEXSET(I)
3 DATA(IGLOB) = 0.0
4 PRD(IGLOB) = .TRUE.
10 CONTINUE
DATA(KGLOBAL) = 1.0
99 RETURN
END

```

```

GO TO 10
C
CC MISSING VALUE OF THE CONDITION ITSELF
C
5 WRITE (6,105) T,I,J,TABNO,IJ
105 FORMAT (1H0,10X,'SUSPENDED EXECUTION OF TABLE ',I3,
1 1X,' AT CONDITION ',I3,' OF RULE ',I3/I11,' STARTED EXECUTION',
2 2X,' OF TABLE ',I3,' TO OBTAIN VALUE OF DATA NUMBER ',I4)
GO TO 10
C
CC MISSING INGREDIENT OF AN ACTION
C
6 WRITE (6,106) T,I,J,DATA,TABNO
106 FORMAT (1H0,10X,'SUSPENDED EXECUTION OF TABLE ',I3,
1 1X,' AT ACTION ',I3,' OF RULE ',I3/I11,' REASON: MISSING ',
2 1X,' INGREDIENT CORRESPONDING TO DATA NUMBER ',I3/
3 31X,' STARTED EXECUTION OF TABLE ',I3)
GO TO 10
C
CC DIRECT EXECUTION
C
7 WRITE (6,107) T,I,J,TABNO
107 FORMAT (1H0,10X,'SUSPENDED EXECUTION OF TABLE ',I3,
1 1X,' AT ACTION ',I3,' OF RULE ',I3/I11,' STARTED EXECUTION',
2 2X,' OF TABLE ',I3,' FOR DIRECT EXECUTION')
10 T = TABNO
C
C IF TABLE T DOES NOT EXIST, TERMINATE EXECUTION.
C
CD WRITE(7,110) L(T)
CD110 FORMAT(1X,'L(T)=',I4)
IF(L(T).EQ.0) GO TO 9999
TABNO = 0
RETURN
9999 WRITE(7,109) T
WRITE(6,109) T
109 FORMAT(1X,'TABLE NUMBER',2X,I4,' DOES NOT EXIST.',
1 1X,' EXECUTION TERMINATED.')
STOP
END
SUBROUTINE OUTPUT(ICYCLE)
C
C THROUGH THIS SUBROUTINE, IT IS POSSIBLE TO OUTPUT ANY DATA
C VALUE FOR CHECKING AND DIAGNOSIS
C
C DECLARATIONS
C
C IMPLICIT LOGICAL*1 (P), INTEGER*2 (I-N)
C INTEGER*2 STACK,ENTRY,T,TABNO,TABD,TABDR,TFIRST,THESET
C COMMON /HICA/DATA,PRD
C DIMENSION DATA(700),PRD(700)
C WRITE (6,100) ICYCLE

```















**B30145**